Project Report

Wenbo Hou
3/19/17
CS496

1. Project Description

My final project is cloud only project whose theme is STEAM User Tracker. My project allows a user to create an account with Google ID as the identifier and STEAM ID as the input data. When creating accounts, users need provide a valid access token generated by using Google OAuth 2.0 API. My project will store users' Google ID and STEAM ID in one of my NDB entities, **Users**.

The NDB entity, **Users** not only contain those two IDs but also information associated with provided steam ID, including active states and game owned by the STEAM ID. The project will invoke two web APIs from STEAM Community to get these information. The project will not store game detail in the NDB entity, **Users**. Instead, it store game details in another NDB entity, **Games**, and store links pointing to owned games in **Users** entity.

All operations in my project requires valid access tokens. My project will detect invalid access token and return error messages. Of course, the project support an arbitrary users, and I will demonstrate it in the video.

2. Key Information

a. Entity Property
    a. Users
        i. Google ID
        ii. Steam ID
        iii. Owned Games
        iv. Active State
    b. Games
        i. Game ID
        ii. Game Name
        iii. User ID (Google ID in Users)
        iv. Total play time (in minutes)
        v. Recent played

b. Entity Relationship
    a. Add users will automatically add games
    b. Each users can have one or more games
    c. Each games must associate with one user
    d. Delete a user will automatically delete games associated with him/her

c. Implemented STEAM API
    a. http://api.steampowered.com/ISteamUser/GetPlayerSummaries/v0002/?key=XXXXXXXXXXXXXXXXXXXXXXXX&steamids=

i. Return users summaries associated with passed STEAM IDs

b. http://api.steampowered.com/IPlayerService/GetOwnedGames/v0001/?key=XXXXXXXXXXXXXXXXX&steamid=76561197960434622&format=json
   i. Return details of games owned by the steam id in json form

d. REST API
   a. POST
      i. https://final-161118.appspot.com/addUser
         1. add a new user with json body containning access token and steam ID
   b. GET
      i. https://final-161118.appspot.com/User?access_token=xxx
         1. Get the user information associate with the access token passed in the URL
      ii. https://final-161118.appspot.com/Game?access_token=xxx
         1. Get games information associate with the access token passed in the URL
      iii. https://final-161118.appspot.com/Game/game_id
         1. A private URL stored in **Users**, entity, used to access a game object.
   c. PUT
      i. https://final-161118.appspot.com/User
         1. A json body will associate with this URL containing the replaced user's id, new Google ID and new STEAM ID
         2. It will delete with that user and delete games associate with the old user. Then, it add new user and new games in NDB
   d. PATCH
      i. https://final-161118.appspot.com/User
         1. A json body will associate with the URL containing an access token and one or more properties value of **Users** entity.
         2. It will update information associate with the user pointed by the access token
   e. DELETE
      i. https://final-161118.appspot.com/deleteUser?access_token=xxx
         1. Delete the user pointed by the passed access token. Game associated with the user will also be deleted