

Holly Harmon
CS 419
Wolford
OSUID: 932456676

Assignment 3: Part 2

URL: <http://api-holly.appspot.com/>

URL Structure:

/user: Post allows for creation of new User entities. Requires a unique "name" value. Get request will return a list of all of the User entity keys.

/user/<id:[0-9]+>/:?: This resource with a Delete request will delete the indicated User entity along with all of its associated BikeEntry and RunEntry entities listed in bikes and runs.

/bike: Post request allows for BikeEntry entity creation. Must include a user key/id and a date. Get will display the keys for all BikeEntry entities.

/bike/<id:[0-9]+>/:?: Get request will return the information for the BikeEntry entity indicated by the id.

/run: Post request allows for RunEntry entity creation. Must include a user key/id and a date. Get will display the keys for all RunEntry entities.

/run/<id:[0-9]+>/:?: Get request will return the information for the RunEntry entity indicated by the id.

/user/<uid:[0-9]+>:/>/bike/<bid:[0-9]+>/:?: A Put request will update the indicated bike entry with the provided data and a Delete request will delete the BikeEntry entity and remove reference to it from the User's bikes[] list.

/user/<uid:[0-9]+>:/>/run/<bid:[0-9]+>/:?: A Put request will update the indicated run entry with the provided data and a Delete request will delete the RunEntry entity and remove reference to it from the User's runs[] list.

Schema Changes:

The biggest change I made to my schema was to have all of the bike and run entries entered into lists as properties, instead of having the User entity be a parent of the BikeEntry and RunEntry entities associated with it. This could have still been done, but the list/repeated properties in the User entities that were demonstrated in the lecture seemed like a more

efficient way to implement it. However, does arranging them the way I did mean that the users and their entries are not in their own entity groups?

After Finishing:

Now that I've finished the API, the first thing I would fix would be the ability to set "kilos" to False when it's already been set to True. It's an easy, but somewhat important fix. The other glaring omission I realized as I was writing up test calls was that the API allows for updates to BikeEntry and RunEntry entities, but not User entities. Once a User is created it can be deleted, but not edited/updated. I think the overall organization could use a little work as well. I included the PUT and DELETE functions for bike and run entries in the user.py section of the program and am wondering if it would have made more sense to include them in entry.py just to keep the user and entry aspects more separate.

I'm also concerned about the data retrieval efficiency because of the changes I made to the schema listed above. This will be something I'll need to look more into or bring up in class discussion, because I didn't think about entity groups and how they work with lists or how much of a difference that makes to overall speed/functionality.