Damian Kulp
Justin
CS496
3/19/2017

Final Assignment – API Documentation

Request Headers

Include the following with every route.

```
Host: https://cs496-157709.appspot.com
Accept: application/json
```

Authentication Routes

Name	Verb	Route	Description
Home Page	GET	/	If signed in: shows name and a sign out button If not signed in: shows date and a sign in button
Sign in	GET	/login	Redirects to the Google auth page
OAuth Callback	GET	/oauth2callback	Receives info from Google, sets cookie, redirects to home
Sign out	GET	/signout	Deletes cookie and redirects home

Task List Routes

Name	Verb	Route	Description
Create a Task List	POST	/tasklists	Create a new task list
Get a Task List	GET	/tasklist/:tasklistId	Get a task list and all its associated tasks by task list id
Delete a Task List	DELETE	/tasklist/:tasklistId	Remove a task list and all its associated tasks by task list id
Patch a Task List	PATCH	/tasklist/:tasklistId	Update an existing task list

Task List Parameters

Name	Parameters	Example

Create a Task List	name – of task list
	description – of task list
	completeBy – date to finish task by
		/tasklist/
		{
		"name": "Make Shoes",
		"description": "Craft leather shoes",
		"completeBy": "3/26/2017"
		}

Get a Task List	tasklistId – datastore id of task list to get	/tasklist/8546312597485312
Delete a Task List	tasklistId – datastore id of task list to remove	/tasklist/2168547932659485

Patch a Task List	tasklistId – datastore id of task list to remove
	name – of task list
	description – of task list
	completeBy – date to finish task by
		/tasklist/1596324877536842
		{
		"name": "Make Shoes",
		"description": "Craft leather shoes",
		"completeBy": "3/26/2017"
		}

Task Routes

Name	Verb	Route	Description
Create a Task	POST	/tasks	Create a new task
Get a Task	GET	/task/:taskId	Get a task by its Datastore id
Delete a Task	DELETE	/task/:taskId	Remove a task in the Datastore
Patch a Task	PATCH	/task/:taskId	Update an existing task

Task Parameters

Name	Parameters	Examples

Create a Task	name – of task
	description – of task
	color – of task background
	taskList – the task should be part of
	weight – of task (importance)
		/tasks/
		{
		"name": "Watch a cricket game",
		"description": "Love me some cricket!",
		"color": "#64AADF",
		"taskList": 1937826459371285,
		"weight": 1
		}

Get a Task	taskId – datastore id of task	/task/9371258641973692
Delete a Task	taskId – datastore id of task	/task/7391682451982736

Patch a Task	taskId – datastore id of task
	name – of task
	description – of task
	color – of task background
	taskList – the task should be part of
	weight – of task (importance)
		/task/3796421573148269
		{
		"name": "Watch a cricket game",
		"description": "Love me some cricket!",
		"color": "#64AADF",
		"taskList": 1937826459371285,
		"weight": 2
		}

Notes

Authentication

All Routes are protected: user must both be signed in and have access to the requested data. Permissions errors will result in the appropriate 40X error.

Related Data

Tasks and Task lists are related, all tasks must have an associated task list. Retrieving or removing a task list will retrieve or remove the associated tasks. Modifying a task list does not modify the associated tasks.

Number of Properties

Though each entity type has only 3 properties in the api, there are more properties stored behind the scenes, these can be retrieved using the appropriate get route.

3rd Party API

I chose to have related data. As such, there is no 3rd party API in the this REST API.

Postman Authentication

I was unable to get Postman to do OAuth the right way. I've included cookies as headers from my personal account and my OSU account. The Postman scripts can be modified to show this ({{osu}} is my OSU account cookie and {{auth}} is my personal cookie) as long as the postman environment file has been imported.