

Account

Name	createNewAccount
Job	Create a new row for a new user in Accounts database
Parameters	username: String email: String password: String type: int
Return	nothing

Name	credentialsMatch
Job	Check if the password that user input is right
Parameters	idtype: "username" or "email", indicates the type of id. id: String, could be an email or a username, depends on idtype. password: String
Return	true, if the credentials match false o\w null, if the username does not exist

Name	usernameForEmail
Job	Return the username with the given email
Parameters	email: String, the email to be searched for
Return	the username of user with email = email, if the user is found null o\w

Name	updateCredentials
Job	Update user's email/password with the new one
Parameters	type: String, either 'email' or 'password' username: String, the user who requested the update newCredential: String, the new email or password, depending on type
Return	nothing

Token

Name	addToken
Job	Create a new row for a new user in Token database
Parameters	token: String, the user's token username: String expirationTime: Number, int
Return	nothing

Name	deleteToken
Job	Delete user's token in the database
Parameters	token: String, the token to be deleted.
Return	nothing

Name	getUsernameByToken
Job	Return the username with the given token
Parameters	token: String, the token to be deleted.
Return	username: String, the username corresponding to the token. null if the token is not found in the database, or has expired.

Profile - MySQL

Name	addUserProfile
Job	Create a new row for a new user in Profile database
Parameters	username: String
Return	nothing

Name	updateProfile
Job	Update user's profile in the Profile database
Parameters	username: String, the user updates: - updates: Object, keys are fields to be updated, values are the new value for the said field
Return	nothing

Name	getProfileByUserName
Job	Return a JavaScript Object where each field contains user's profile info in the database
Parameters	username: String
Return	An object. Keys are each field. Values are the data in the database.(use 'undefined' or 'NaN' to check if the data exist)

Name	getUserByType
Job	Return the type of the user
Parameters	username: String
Return	A string indicating user's type

Community - Neo4j

Name	createUser
Job	Create a user node in the database
Parameters	username: String
Return	nothing

Name	makePost
Job	Create a postNode with given data and create 'CREATE_POST' relationship between user and post
Parameters	username: String content: String, the content of the post title: String, the title of the post postId: the unique postId of the post time: the time that user makes the post
Return	nothing

Name	deletePost
Job	Delete the post node and any relationship with this post node (However, related reply will not be deleted)
Parameters	postId: the postId of the post that user reply to
Return	nothing

Name	searchPostByTitle
Job	Return a set of objects where each object contains postId, title and time
Parameters	title: String, part of the string of the title
Return	A set of object looks like: [{postId: A01, title: new_world, time: 2000-01-01, content: "Hi"}, {postId: B09, title: hello_world, time: 2020-10-10, content: "Hello"}]

Name	getPosts
Job	Return a set of objects where each object contains postId, title and time sorted by time, and returns posts numbers skip = pageNum * postPerPage limit = postPerPage
Parameters	pageNum: int postsPerPage:int
Return	A set of object looks like: [{postId: A01, title: new_world, time: 2000-01-01, content: "Hi"}, {postId: B09, title: hello_world, time: 2020-10-10, content: "Hello"}]

Name	getComments
Job	Return a set of objects that are comments of the original post where each object contains postId, title and time sorted by time
Parameters	postId: String
Return	An object filled with comments under the original post

Name	searchPostById
Job	Return a set of objects where each object contains postId, title and time
Parameters	Id: Part of the target postId
Return	A set of object where each object contains postId, title, time and content (similar to the above method)

Name	searchPostByUser
Job	Return a set of objects where each object contains postId, title and time. Each object will be the post that user created
Parameters	Username: the username of the user
Return	A set of object where each object contains postId, title, time and content (similar to the above method)

Name	makeReply
Job	Create a reply node, 'CREATE_REPLY' relation between user and the reply and 'REPLY_TO' relation between the post/reply and the reply
Parameters	username: String content: String, the content of the Reply replyId: the unique id of the reply targetId: the Id of the target that user reply to time: the time that user makes the reply type: the type of the target (should be "post" or "reply")
Return	nothing

Name	deleteReply
Job	Delete the reply, 'CREATE_REPLY' and any 'REPLY_TO' relation with this reply
Parameters	replyId: the unique Id of the reply
Return	nothing

Name	addTag
Job	Create a tag and the 'HAS_TAG' relationship between the user and tag. If the tag exists in the database (has the same tagName), then only the "HAS_TAG" relationship will be created.
Parameters	username: String tagName: String
Return	nothing

Name	getTags
Job	Gets all the tags that a user has.
Parameters	username: String
Return	An object containing all of the tags that the user has.

Name	userHasTag
Job	Checks if the user has the given tag
Parameters	username: String tagName: String
Return	True if the user has a 'HAS_TAG' relationship with the tag, False if the user does not.

Name	deleteTag
Job	Delete the relationship between the user and the tag, and the tag node if there are no relationships involving the tag
Parameters	username: String tagName: String
Return	nothing

Name	followPost
Job	Create 'FOLLOW' relationship between the user and the post
Parameters	username: String postId: String, the postId of the post that user wants to follow
Return	nothing

Name	unfollowPost
Job	Delete 'FOLLOW' relationship between the user and the post
Parameters	username: String postId: String, the postId of the post that user wants to unfollow
Return	nothing

Name	getFollowedPostByUser
Job	Return a set of postId where each post is followed by the user
Parameters	username: String
Return	A set of postId that user follows Empty if user follows nothing

Name	getFollowingUserByPost
Job	Return a set of username where each user is following this post
Parameters	postId
Return	A set of username where each user is following this post Empty if user follows nothing