

## Use Cases & Corresponding API Calls

1. User creates an account for the first time
  - Firebase authentication handles sign up with email.
  - Frontend makes a call to *check\_email* to see if a *user\_id* is associated with the email account the user authenticated with. Since this is a new account, there should be no binding.
  - Frontend then asks the user for a unique *user\_id*.
  - Frontend makes a call to *search* to see if the *user\_id* is taken. If taken, keep asking user to enter a user id and check until they enter something that is not already taken.
  - Frontend continues asks for account information, forms a JSON packet with the user credentials.
  - Frontend makes a call to *create\_user* with user credentials attached.
2. User logs in
  - Firebase authentication handles sign in with email.
  - Frontend makes a call to *check\_email* to fetch the associated *userId*.
  - Frontend uses the *userId* to make calls to *get\_my\_feeds* and *get\_my\_posts* to gather the content to build the user's wall.
  - Frontend renders user wall.
3. User searches for a friend
  - Frontend makes a call to *is\_friend* to see if the *userId* provided is already a friend of the user.
  - Frontend makes a call to *search* to see if the *userId* is in our system.
  - If there is such a user and the user is already friends with us, we may show some more information about the user.
  - If there is such a user and the user is not friends yet, frontend will display option to send a request.
4. User adds a friend
  - Continuing from where a user searches for another user, Frontend makes a call to *friend\_request* to send a friend request to the other user.
  - On the frontend of the other user, a friend request appears when we refresh and a call to *get\_pending\_tasks* was made inadvertently through refresh.
  - If the other user accepts the request, Frontend makes a call to *friend\_response*.
  - Back to the frontend of the befriending user, as soon as friend response appears when we refresh and a call to *get\_pending\_tasks* was made inadvertently through refresh, the Frontend makes calls to *add\_friend\_userid* for both users to officially record them as friends.

5. User deletes a friend
  - Frontend makes a call to *delete\_friend\_userid*.
6. User creates a post
  - Frontend makes a call to *create\_post*.
7. User updates a post
  - Frontend makes a call to *update\_post*.
8. User deletes a post
  - Frontend makes a call to *delete\_post*.
9. User comments on a post
  - Frontend makes a call to *create\_comment*.
10. User updates a comment
  - Frontend makes a call to *update\_comment*.
11. User deletes a comment
  - Frontend makes a call to *delete\_comment*.
12. User replies to a comment
  - Frontend makes a call to *create\_reply*.
13. User updates a reply
  - Frontend makes a call to *update\_reply*.
14. User deletes a reply
  - Frontend makes a call to *delete\_reply*.
15. User clicks a friend's profile to view the friend's posts
  - Frontend makes a call to *get\_friendposts*.
16. User clicks on his own profile to see his/her homepage
  - Frontend makes a call to *get\_myposts*.
17. User refreshes his page to get the most recent feed
  - Frontend makes a call to *get\_my\_feed*.
  - Frontend makes a call to *get\_pending\_tasks* to deal with friend requests or friend responses.

## API Documentation: RIGHT NOW NO AUTHENTICATION

#####

### User Related APIs

#### - **create\_userid: creates a new user in database**

```
/*
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/create_userid
* params: user profile: JSON
* return:
* if user doesn't exist:    200:    return "user created!" with status code: 1
* if user exists:          200:    return JSON of existing user profile with status code 0
* if add fails:            error:   return error code
*/
```

#### - **delete\_userid: delete the specified user in database**

```
/*
* method: DELETE
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/delete_userid
* params: JSON containing user_id
* return:
* if user exists:          200:    return "user deleted!" with status code: 1
* if user doesn't exist:   200:    return "no such user" with status code: 0
* if add fails:            error:   return error code
*/
```

#### - **update\_userid: updates the specified user profile in database**

```
/*
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/update_userid
* params: user profile JSON
* return:
```

\* if user does exist: 200: return true with status code: 1  
\* if user doesn't exist: 200: return false with status code: 0  
\* if add fails: error: return error code

\*/

- **search\_userid: search for the specified user in database**

/\*

\* method: GET

\* url:

https://us-central1-cse112sp18teamxii.cloudfunctions.net/search\_userid?user\_id={id}

\* params: user\_id

\* return:

\* if user exists: 200: return true with status code: 1  
\* if user doesn't exist: 200: return false with status code: 0  
\* if search fails: error: return JSON of user profile with error code

\*/

#####  
**Friend Related APIs**

- **is\_friend: check if two users are friends**

/\*

\* method: GET

\* url:

https://us-central1-cse112sp18teamxii.cloudfunctions.net/isFriend?user\_id={id}&me\_id={id}

\* params: user\_id

\* return:

\* if user exists: 200: return whether friends(true/false) with status code: 1  
\* if user doesn't exist: 200: return false with status code: 0  
\* if search fails: error: return error code

\*/

- **friend\_request: me\_id sends user\_id a friend request (not friends yet)**

/\*

```

* method: GET
* url:
https://us-central1-cse112sp18teamxii.cloudfunctions.net/friend_request?user_id={id}&
me_id={id}
* params: user_id & me_id
* return:
* if user exists:          200:  return true with status code: 1
* if user doesn't exist:   200:  return false with status code: 0
* if add fails:            error: return error code
*/

```

- **friend\_response: me\_id sends user\_id a friend response(upon seeing friend request)**

```

/*
* method: GET
* url:
https://us-central1-cse112sp18teamxii.cloudfunctions.net/friend_response?user_id={id}
&me_id={id}
* params: user_id & me_id
* return:
* if user exists:          200:  return true with status code: 1
* if user doesn't exist:   200:  return false with status code: 0
* if add fails:            error: return error code
*/

```

- **add\_friend\_userid: me\_id & user\_id becomes friend(upon seeing friend response)**

```

/*
* method: GET
* url:
https://us-central1-cse112sp18teamxii.cloudfunctions.net/add_friend_userid?user_id={id}
}&me_id={id}
* params: user_id & me_id

```

```

* return:
* if user exists:          200:  return "success" / "already friends" with status code: 1
* if user doesn't exist:   200:  return "no such user/me" with status code: 0
* if add fails:            error: return error code
*/

```

**- delete\_friend\_userid: user\_id and me\_id becomes strangers**

```

/*
* method: DELETE
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/delete_friend_userid
* params: JSON containing user_id & me_id
* return:
* if user exists:          200:  return "friendship ends!" or "you're not friends!" with
status code: 1
* if user doesn't exist:   200:  return "no such user" or "no me" with status code: 0
* if group doesn't exist:  200:  return "no such group" with status code: 0
* if add fails:            error: return error code
*/

```

**- clear\_friend\_request: clears all user\_id's friend requests**

```

/*
* method: GET
* url:
https://us-central1-cse112sp18teamxii.cloudfunctions.net/clear_friend_request?user_id=
{id}
* params: user_id
* return:
* if user exists:          200:  return true with status code: 1
* if user doesn't exist:   200:  return false with status code: 0
* if add fails:            error: return error code
*/

```

**- clear\_friend\_response: clears all user\_id's friend response**

```

/*
* method: GET
* url:
https://us-central1-cse112sp18teamxii.cloudfunctions.net/clear_friend_response?user_id={id}
* params: user_id
* return:
* if user exists:          200:  return true with status code: 1
* if user doesn't exist:   200:  return false with status code: 0
* if add fails:            error: return error code
*/

```

**- get\_friend\_request: get user\_id's friend requests**

```

/*
* method: GET
* url:
https://us-central1-cse112sp18teamxii.cloudfunctions.net/get_friend_request?user_id={id}
* params: user_id
* return:
* if user exists:          200:  return true with status code: 1
* if user doesn't exist:   200:  return false with status code: 0
* if add fails:            error: return error code
*/

```

**- get\_friend\_response: get user\_id's friend response**

```

/*
* method: GET
* url:
https://us-central1-cse112sp18teamxii.cloudfunctions.net/get_friend_response?user_id={id}
* params: user_id

```

```
* return:
* if user exists:          200:  return true with status code: 1
* if user doesn't exist:   200:  return false with status code: 0
* if add fails:            error: return error code
```

```
*/
```

```
#####
```

### **Group Related APIs:**

**- set\_group: set the group\_id for multiple friends**

```
/*
```

```
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/set_group
* params: user_ids(array of user_id), me_id, group_id in JSON
* return:
* if each user is a friend of me:  set is true(regardless of user current group)
* if not:                          return false and code 0
* else:                            return error code
```

```
*/
```

**- unset\_group: release multiple friends from the same group**

```
/*
```

```
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/unset_group
* params: user_ids, me_id, group_id
* return:
* if each user is a friend of me:  unset is true (regardless of user's current group)
* if not:                          return false and code 0
* else:                            return error code
```

```
*/
```

```
#####
```

### **Post Related APIs**

**- create\_post: when user creates a new post**

```
/*
```



```

* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/create_post
* params: post profile JSON: requires user_id; group_id is defaulted "0" if not specified
* return:
* if create post is successful:    return post_id in posts(collection) with code 1
* if user_id/group_id is not valid: return "no such user" or "no such group" with code 0
* if not:                          return error code
*/

```

**- update\_post: when user updates his/her post**

```

/*
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/update_post
* params: post profile JSON: requires user_id & group_id
* return:
* if update post is successful:    return "post updated!" or "this post is not yours!"
with code 1
* if no such post/user:           return "no such post/user" with code 0
* if not:                          return error code
*/

```

**- delete\_post: when user deletes his/her post**

```

/*
* method: DELETE
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/delete_post
* params: post profile JSON
* return:
* if delete post is successful:    return "post deleted!" or "this post is not yours!"
with code 1
* if post/user is invalid:         return "no such post/user" with code 0
* if not:                          return error code
*/

```

- **get\_mypost: when user visits his/her homepage**

```
/*  
* method: GET  
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/get_mypost?user_id={id}  
* params: user_id  
* return:  
* if get is successful:      return user's posts (no comments/replies) with code 1  
* if user doesn't exist:    return {} with code 0  
* if not:                   return error code  
*/
```

- **get\_friendpost: when user visits a friend's homepage**

```
/*  
* method: GET  
* url:  
https://us-central1-cse112sp18teamxii.cloudfunctions.net/get_friendpost?user_id={id}&friend_id={id}  
* params: user_id, me_id  
* return:  
* if get is successful:      return friend's posts with code 1  
* if user doesn't exist or not friends:    return {} with code 0  
* if not:                   return error code  
*/
```

- **get\_myfeed: when user visits his/her main page**

```
/*  
* method: GET  
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/get_myfeed?user_id={id}  
* params: user_id  
* return:  
* if get is successful:      return user's feeds with code 1  
* if user doesn't exist:    return {} with code 0
```

```

* if not:                return error code
*/

    - like_post: when user likes a post
/*
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/like_post
* params: user_id & post_id
* return:
* if like post is successful:    return true/false if user not liked/liked before
* if not such user:             return false
* if not:                       return error code
*/

    - unlike_post: when user unlikes a post
/*
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/unlike_post
* params: user_id & post_id
* return:
* if like post is successful:    return true/false if user liked/unliked before
* if not such user:             return false
* if not:                       return error code
*/

#####
Comment Related APIs

    - create_comment:
/*
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/create_comment
* params: post_id,user_id, comment JSON
* return:

```

```

* if comment is successful:      return comment_id with code 1
* if invalid user/post:         return "no such user" or "no such post" with code 0
* if else:                      return error
*/

    - update_comment
/*
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/update_comment
* params: post_id,user_id, comment_id,comment JSON
* return:
* if update is successful:      return comment_id with code 1
* if invalid user/post/comment_id: return "no such user/post/comment!" with 0
* if else:                     return error
*/

    - delete_comment
/*
* method: DELETE
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/delete_comment
* params: post_id,user_id, comment_id,comment JSON
* return:
* if delete is successful:      return "success" with code 1
* if invalid user/post/comment: return "no such post/user/comment!" with code 0
* if else:                     return error
*/

    - show_comment
/*
* method: GET
* url:
https://us-central1-cse112sp18teamxii.cloudfunctions.net/show_comment?post_id={id}
* params: post_id

```

```

* return:
* if show is successful:    return JSON of all comments (no replies)
* if invalid user/post:    return "no such postr!" with code 0
* if else:                 return error
*/

```

## ##### Reply Related APIs

### - create\_reply

```

/*
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/create_reply
* params: post_id,user_id, comment_id, reply JSON
* return:
* if reply is successful:    return reply_id
* if invalid user/post:    return "no such user/comment!" code 0
* if else:                 return error
*/

```

### - update\_reply

```

/*
* method: POST
* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/update_reply
* params: post_id,user_id, comment_id,reply_id, reply JSON
* return:
* if update is successful:    return reply_id
* if invalid user/post:    return "no such user/comment/post/reply!" with code 0
* if else:                 return error
*/

```

### - delete\_reply

```

/*
* method: DELETE

```

```

* url: https://us-central1-cse112sp18teamxii.cloudfunctions.net/delete_reply
* params: post_id,user_id, comment_id,reply_id, reply JSON
* return:
* if delete is successful:    return "success"
* if invalid user/post:      return "no such user/comment/post/reply!" with code 0
* if else:                   return error
*/

- show_reply
/*
* method: GET
* url:
https://us-central1-cse112sp18teamxii.cloudfunctions.net/show_reply?post_id={id}&com
ment_id
* params: post_id,comment_id
* return:
* if show is successful: return JSON of all replies
* if invalid user/post: return "no such comment/post!" code 0
* if else:               return error
*/

```