IF A REQUEST RETURNED A STATUS CODE OF **500**, THEN THERE IS SOMETHING WRONG WITH THE BACK END!

The posts database

General URL: https://us-central1-comp426-firebase.cloudfunctions.net/posts

Post New Posts

```
const result = await axios({
        method: 'post',
        url: 'https://us-central1-comp426-firebase.cloudfunctions.net/posts',
        data:{
           "body": "What ever the post body message is",
           "uid": The User ID who posted this post,
           "userName": "kmp",
           "anonymous": true
           "postTo": The User ID of who this post is to.
        }
});
```

The comment array and the liked array will be auto-generated and set to [].
There will also be a createdAt time generated automatically.
If anything is wrong with the request body, you will get a status code of **400** and a message telling you which error it is. It can either be the required field is not provided or the required field has a value of null.
It will return the ID generated by firebase. (this is going to look like someone smashed the keyboard)

Get Posts

```
const result = await axios({
            method: 'get',
            url: 'https://us-central1-comp426-firebase.cloudfunctions.net/posts'
});
```

This will give back all posts in an array, ordered by the created time (not updated time!!) in descending order (newest on top).

Get Posts By ID

```
const result = await axios({
        method: 'get',
        url: 'https://us-central1-comp426-firebase.cloudfunctions.net/posts/${id}}'
});
```

This will give back the post that matches the post ID. If the ID does not exist, you will receive a status code of **400** and a message telling you so.

Delete Posts By ID

```
const result = await axios({
        method: 'delete',
        url: 'https://us-central1-comp426-firebase.cloudfunctions.net/posts/${id}}'
});
```

- This will give back true if the post according to the ID is deleted. If the ID does not exist, you will receive a status code of **400** and a message telling you so.

Update/Put Posts By ID

```
const result = await axios({
        method: put,
        url: 'https://us-central1-comp426-firebase.cloudfunctions.net/posts/${id}}'
        data:{
                "body": "Whatever the post body message is",
                "uid": The user ID that this post is created from,
                "userName": "kmp",
                "anonymous": true,
                "comments": [],
                "likes": []
                "postTo": User ID that the post is for.
      }
});
```

This will return true if the update is successful.
You will receive a status code of 400 if any of the required fields above is missing or has a value of null or "".

The Users database

Create/Post new Users

```
const result = await axios({
        method: 'post',
        url: 'https://us-central1-comp426-firebase.cloudfunctions.net/users',
        data:{
                "userName" : "Some Username"
                "email": "someEmailAddress"  // The server does not check if this is a real email address. The front end will have to do this!
                "password": "whatever the password is"
        }
});
```

- The highestGameScore, matchPoint, and friends fields will all be auto-generated when creating a new user. The highestGameScore will be set to 0, the friends will be set to an empty array, and the matchPoint array will be and an array of five zeros([0,0,0,0,0]).
- If anything is wrong with the request body, you will get a status code of **400** and a message telling you which error it is. It can either be the required field is not provided or the required field has a value of null.
- It will return the ID generated by firebase. (this is going to look like someone smashed the keyboard)

Get Users

```
const result = await axios({
            method: 'get',
            url: ‘https://us-central1-comp426-firebase.cloudfunctions.net/users'
});
```

- This will give back all users in an array, ordered by the userName in ascending order.

Get Users By ID

```
const result = await axios({
            method: 'get',
            url: 'https://us-central1-comp426-firebase.cloudfunctions.net/users/${id}}'
});
```

- This will give back the user that matches the user ID. If the user ID does not exist, you will receive a status code of **400** and a message telling you so.

Delete User By ID (If we want to have a button for the users to delete their account? But most likely not doing it)

```
const result = await axios({
        method: 'delete',
        url: 'https://us-central1-comp426-firebase.cloudfunctions.net/users/${id}}'
});
```

- This will give back true if the user according to the ID is deleted. If the ID does not exist, you will receive a status code of **400** and a message telling you so.

Update/Put Users By ID

```
const result = await axios({
        method: put,
        url: 'https://us-central1-comp426-firebase.cloudfunctions.net/users/${id}}'
        data:{
                "userName": "some User name",
                 "email" : "some email address",
                 "password": "some password",
                 "matchPoint": [#,#,#,#,#],
                 "friends": [array of friends user id],
                 "highestGameScore": ##,

     }
});
```

- This will return true if the update is successful.
- You will receive a status code of **400** if any of the required fields above is missing or has a value of null or "".