

Firebase functions written by Stavva and Josh

Called using:

```
firebase.functions().httpsCallable(<function-name-as-a-string>)(<data>)
```

Functions are asynchronous, so a result will need to be awaited

Users

Function Name	Description	Expected Data	Expected Result	Errors
user-add	Adds a new user to the database, the user must have already been created in auth. All properties are required. Email must be a valid email	{ "userId": string, "name": string, "bio": string, "email":validEmail }	{ result: "User \${username} has been added" }	"Either the email is wrong or the userId already exist"
user-delete	Deletes the user with corresponding userId	{ "userId": string }	"deleted user \${userId}"	
user-getOne	Returns a user with the given userId	{ "userId": string }	{ "userId": string, "name": string, "bio": string, "email":validEmail }	
user-update	Updates the user with the given userId with any other properties provided, userId is	{ "userId": string, ... }	{	

	required. Any properties not in the sent data remain untouched.		"result": "User \${userName} has been added" }	
--	---	--	---	--

Projects

Function Name	Description	Expected Data	Expected Result	Errors
project-add	Adds a new project to the database, must have a projectName and userId	{ "userId": string, "projectName": string, "projectDesc": string, "projectBody": string, "projectTags": [string], ... }	{ userId: string projectId: string }	
project-delete	Deletes a project, requires a projectId	{ "projectId": string }	{ message: `Project projectId: \${data.projectId} deleted` }	{ message: `Project projectId: \${data.projectId} could not be deleted`, error: 'delete_error' }
project-getByUser	Returns a list of projects data with the given userId	{ "userId": string }	[{ "userId": string, "projectName": string,	

			<pre>"projectDesc": string, "projectBody": string, "projectTags": [string], ... }, ...]</pre>	
project-getOne	Returns a project with the given projectId	<pre>{ "projectId": string }</pre>	<pre>{ "userId": string, "projectName": string, "projectDesc": string, "projectBody": string, "projectTags": [string], ... }</pre>	<pre>{error: `No matching documents with projectId: \${projectId}`, not_found: true }</pre>
project-update	Updates the project with the given projectId with any other properties provided, projectId is required. userId cannot be modified and attempting to do so returns an error. Any properties not in the sent data remain untouched.	<pre>{ "projectId": string, ... }</pre>	<pre>{ "result": "Project projectId: \${this.projectId} updated" }</pre>	