

Table of Contents for functions/ folder of the peterbook repo

functions/src/api/	3
emailAPI.ts	3
functions/src/data	3
httpStatusCode.ts	4
functions/src/domain/authorize	5
Verification.ts	5
loginUser.ts	6
oauthType.ts	6
registerUserResults.ts	7
userStateTypes.ts	8
functions/src/domain/circles	9
circle.ts	9
userTie.ts	10
functions/src/domain/comments	11
comments.ts	11
functions/src/domain/common	12
baseDomain.ts	12
email.ts	12
feed.ts	12
socialError.ts	12
functions/src/domain/graphs	13
graphs.ts	13
functions/src/domain/imageGallery	14
image.ts	14
functions/src/domain/notifications	15
notification.ts	15
functions/src/domain/posts	16
post.ts	16
functions/src/domain/users	17
profile.ts	17
functions/src/domain/votes	19
votes.ts	19
functions/src/services/authorize	20

authorizeService.ts	20
functions/src/services/comments	21
commentService.ts	21
functions/src/services/common	22
mailService.ts	22
functions/src/services/posts	23
postService.ts	23
functions/src/services/users	24
userService.ts	24

[functions/src/api/](#)

[emailAPI.ts](#)

File Description: This file is used to send email to confirm a user's requested subscription service.

Functions:

Const sendEmail

Parameter: email of Email object from ../domain/common/email

Description: This function is used to send email to confirm a user's requested subscription service.

Returns: A promise that represents completion of sending email or failure in case where it does not.

[functions/src/data](#)

httpStatusCode.ts

File Description: This file is simply an enumeration of HTTP status codes.

functions/src/domain/authorize

Verification.ts

File Description: This file defines a class 'Verification' that meant to be used for authentication and login.

class Verification()

Class Description: This class is to be used for authentication and login.

Attributes/Constructor:

id – a string representing the user's id, given to them (ie assigned user id)

code – a string representing login code

target – a string representing which login or authentication

creationDate – a number representing when a user's account was created

remoteIpAddress – a string representing the remoteIpAddress to be used for authentication servers

userId – a string representing the user's id, given to us (ie a login)

isVerified – a boolean representing if the user's account is verified (valid email)

loginUser.ts

File Description: This file defines a class “LoginUser”

Class:

class LoginUser()

Class Description: This class is used to define a user and their attributes.

Attributes/Constructor:

uid – a string denoting the user’s id

emailVerified – a Boolean defining if the user has a valid email address

providerId – a string denoting which login authentication they used: Facebook, Google, etc

email – the users email

avatarurl – a string that is the url to the user’s avatar

Functions:

emailVerified()

Function Description: Returns a boolean depending if the user’s email is verified or not.

Parameters: None

Returns: a Boolean

providerId()

Function Description: Returns a string denoting the login provider id.

Parameters: None

Returns: A string

displayName()

Function Description: Returns a string denoting the display name for the user.

Parameters: None

Returns: A string

email()

Function Description: Returns a string that is the user’s email.

Parameters: None

Returns: A string

avatarURL()

Function Description: Returns a string that is the user’s url to their avatar

Parameters: None

Returns: A string

oauthType.ts

File Description: This file describes an enumeration of OAuthType.

registerUserResults.ts

File Description: This file describes a class.

Class:

RegisterUserResult

Class Description: This class describes the registration status of a user.

Attributes/Constructor:

uid - a string that is the users id

Functions:

uid()

Function Description: This function returns the uid of RegisterUserResult

Parameters: None

Returns: A string that is the uid of RegisterUserResult

[userStateTypes.ts](#)

File Description: This file describes an enumeration for the types of user states.

functions/src/domain/circles

circle.ts

File Description:

This file describes the friends circle for the social network

Class:

Circle

Class Description:

This class describes the friend circle for the social network.

Attributes/Constructor:

id, string the identifier

Functions:

creationDate

Function Description:

This returns the creationDate for that circle.

Parameters: None

Returns: Date, the creation date

ownerId

Function Description:

This returns the ownerId of that circle.

Parameters: None

Returns: string, the owner's id

Name

Function Description:

This returns the name of that circle.

Parameters: None

Returns: string, the circle's name

isSystem

Function Description:

This returns true if the circle was configured by the system.

Returns: Boolean, true when the circle was configured by the system.

[userTie.ts](#)

File Description:

This file describes the class UserTie

Class:

Class:

UserTie

Class Description: This class describes the user within the circle social network.

Attributes/Constructor:

userId – string for userID

creationDate – Date for the date of the user creation

fullName – string for the full name of the user

avatar – string, the url address for the user's avatar

approved – Boolean, if the user has been approved in the circle.

functions/src/domain/comments

comments.ts

File Description: This file describes the class Comment.

Class:

Comment

Attributes/Constructor:

id: string, the comment identifier

postId (Required): string, the post identifier that comment belong to

text: string, the comment text

score: number, the comment score

creationDate: number, the creation date of the comment

userDisplayName: string, the comment owner's full name

userAvatar: string, the comment owner's url to their avatar

userId: string, the comment owner's identifier

editorStatus: Boolean whether if the comment is in an editable state or not

functions/src/domain/common

baseDomain.ts

File Description: An empty file that is meant to be a layer such that functions are connected to one another, even from different Class.

email.ts

File Description: This file describes the class, Email

Class: Email

Class Description: This represents an email.

Attributes/Constructor:

from : (Required) string, the email's sender

to : (Required) string, the email's recipient

subject : (Required) string, the email's subject

html : string, the email's html

text : string, the email's body

feed.ts

File Description: This file describes the class Feed.

Class:

Feed

Class Description: This class describes the user's feedback

Attributes/Constructor:

id – string, the feedback identifier

text – string, the feedback text

feedType – FeedType, the type of feedback

user – User, the user who requested feed

feedType.ts

File Description: This file describes an enumeration of the type of feedback we receive from the user.

socialError.ts

File Description:

functions/src/domain/graphs

graphs.ts

File Description:

This file describes a class Graph that shows your relationships with other people on your social network.

Class:

Graph

Class Description:

This class describes relationships with other people on your social network.

Attributes/Constructor:

leftNode : (Required) string, represents left node of the graph

edgeType : (Required) string, represents the graph relationship type

rightNode : (Required) string, represents right node of the graph

LeftMetaData : (Required) any, represents the left node's metadata

rightMetadata : (Required) string, represents the right node's metadata

graphMetadata : (Required) string, represents the graph's metadata

nodeId : string, represents the graph node's identifier

functions/src/domain/imageGallery

image.ts

File Description: This file describes the class Image such that the social network can post, comment, share images.

Class:

Image

Class Description: This class describes and allows a means for an image to be posted, commented, and shared.

Attributes/Constructor:

id : (Required) string, the images identifier

creationDate : (Required) number, the images creation date

deleteDate : (Required) string, the images delete date

URL : (Required) string, the image's URL

fullPath : (Required) string, the image folder name with image name {folderName/imageName}

ownerUserId : (Required) string, the image's owner's identifier

lastEditDate : (Required) number, the image's last edit date

deleted: (Required) Boolean, whether or not the image was deleted

functions/src/domain/notifications

notification.ts

File Description: This file describes the class Notification that allows the social network to display notifications.

Class:

Notification

Class Description: This class allows the social network to display notifications.

Attributes/Constructor:

id : string, the notification identifier

description: (Required) string, the description of the notification

url : (Required) string, the URL which notification refers to

notifierUserId : (Required) string, the identifier of the user who makes the notification

notifyReceiverUserId : (Required) string, the identifier of the user who receives the notification

isSeen : (Required) Boolean, if the notification has been seen (true) or (false)

functions/src/domain/posts

post.ts

File Description: This file describes the class Post.

Class:

Post

Class Description: This class details the attributes of a post on the social network.

Attributes/Constructor:

id: string, the post's identifier

postTypeId : number, the identifier of post type

creationDate : number, the date of the post's creation

deleteDate : number, the date of the post's deletion

score : number, the score of the post

votes : an array of array of voterId and a Boolean, lists of voter identifier

viewCount : number, the number of view a post has

body : string, the text of the post

ownerUserId : string, the identifier of post owner

ownerDisplayName : string, the full name of the post owner

ownerAvatar : string, the avatar address (url) of the post's owner

lastEditDate : number, the date of the post's last edit

tags : string array, the number of tags the post has

commentCounter: number, the number of comments a post has

image : string, the url address of the image

imageFullPath : string, the image's full path

video : string, the url address of a post's video

disableComments : Boolean, if writing a comment is disabled (True) or not (False)

disableSharing : Boolean, if sharing the post is disable (True) or not (False)

deleted : Boolean, if the post is deleted (True) or not (False)

functions/src/domain/users

[profile.ts](#)

File Description: This file describes the class Profile

Class:

Profile

Class Description: This class describes the user's profile.

Attributes/Constructor:

avatar : (Required) string, the url to the user's avatar

fullName : (Required) string, the user's full name

banner : (Required) string, the url to the user's banner

tagLine : (Required) string, the user's tagline for their page

creationDate : (Required) number, the date of which the user's profile was created

email : string, the email the user listed on their profile

[user.ts](#)

File Description: This file describes the class User

Class:

User

Class Description: This class describes what is a user on the social network.

Attributes/Constructor:

fullName : (Required) string, the user's full name

avatar : (Required) string, the url to the user's address

email : string, the user's email

password : string, the user's password

userId : string, the user's identifier

creationDate : string, the user's date of account creation

[userProvider.ts](#)

File Description: This class describes the class UserProvider

Class:

UserProvider

Class Description: This class describes which user used to provide as a login.

Attributes/Constructor:

userId : (Required) string, the user's identifier

email : (Required) string, the user's email

fullName : (Required) string, the user's full name

avatar : (Required) string , the user's avatar

providerId : (Required) string, the user's provider identifier (id for Facebook, github,etc)

provider : (Required) string, the user's provider (Facebook, Github, etc)

functions/src/domain/votes

votes.ts

File Description: This file describes the class Vote.

class

Vote

Class Description: This class describes how voting is made on the social network.

Attributes/Constructor:

postId : (Required) string, post identifier which vote on

creationDate : (Required) number, the vote date

userDisplayName : (Required) string, voter's full name

userAvatar : (Required) string, url address of the user's avatar

userId : (Required) string, the voter's identifier

[functions/src/services/authorize](#)

[authorizeService.ts](#)

File Description: This file implements on user create, a middle ware that validates Firebase ID tokens passed in the Authorization HTTP header, SMS verification, and user registration.

[publicAuthService.ts](#)

File Description: This file implements login user API, email verification, and password reset API.

[functions/src/services/comments](#)

[commentService.ts](#)

File Description: This file implements adding and deleting comments.

[functions/src/services/common](#)

[mailService.ts](#)

File Description: This file implements mailService within the social network.

`functions/src/services/posts`

`postService.ts`

File Description: This implements posting/routing service for posts on the social network.

[functions/src/services/users](#)

[userService.ts](#)

File Description: This file implements user services like getting users by user identifier list, getting users by http trigger, routing posts, and handle on update for user information