

# GODSocialNetIO

Eduardo Alexandre, Aline Borges, Leonardo Haddad, Thiago Araujo

December 9, 2014

## Contents

<b>1</b>	<b>GODSocialNetIO</b>	<b>1</b>
1.1	Team . . . . .	1
1.2	Description . . . . .	1
1.3	Class Diagram . . . . .	2
1.4	Documentation . . . . .	4
1.4.1	Fetchers . . . . .	4
1.4.2	GODDdatas . . . . .	10
1.5	Contact . . . . .	28

## 1 GODSocialNetIO

### 1.1 Team

Eduardo Alexandre, Aline Borges, Leonardo Haddad, Thiago Araujo

### 1.2 Description

Our module for the GOD Project is to allow others modules to communicate with the Facebook and Twitter servers. It does that by implementing both public APIs.

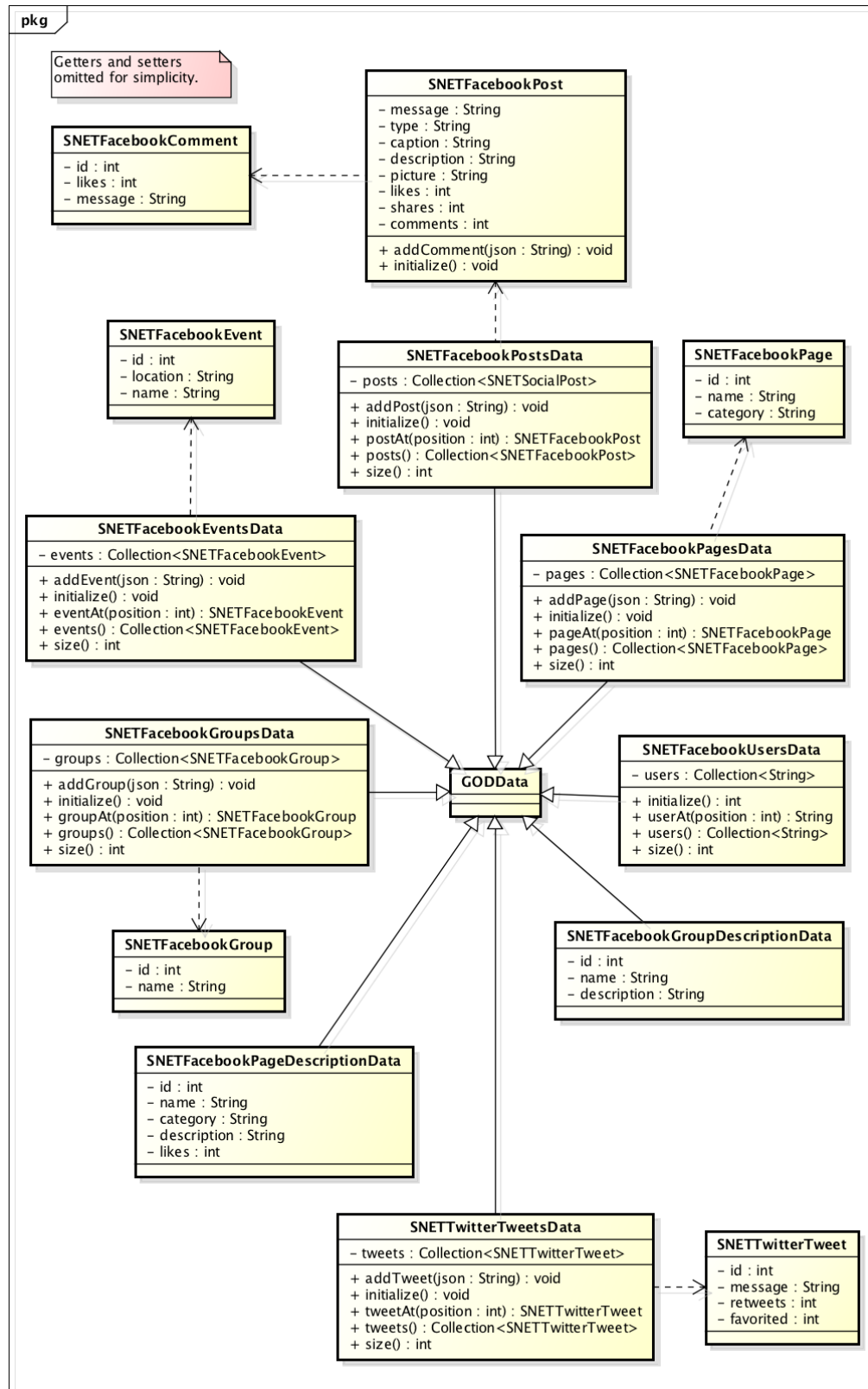
The module is divided in two main parts, first are the fetchers, these classes are responsible to all the communication with the APIs.

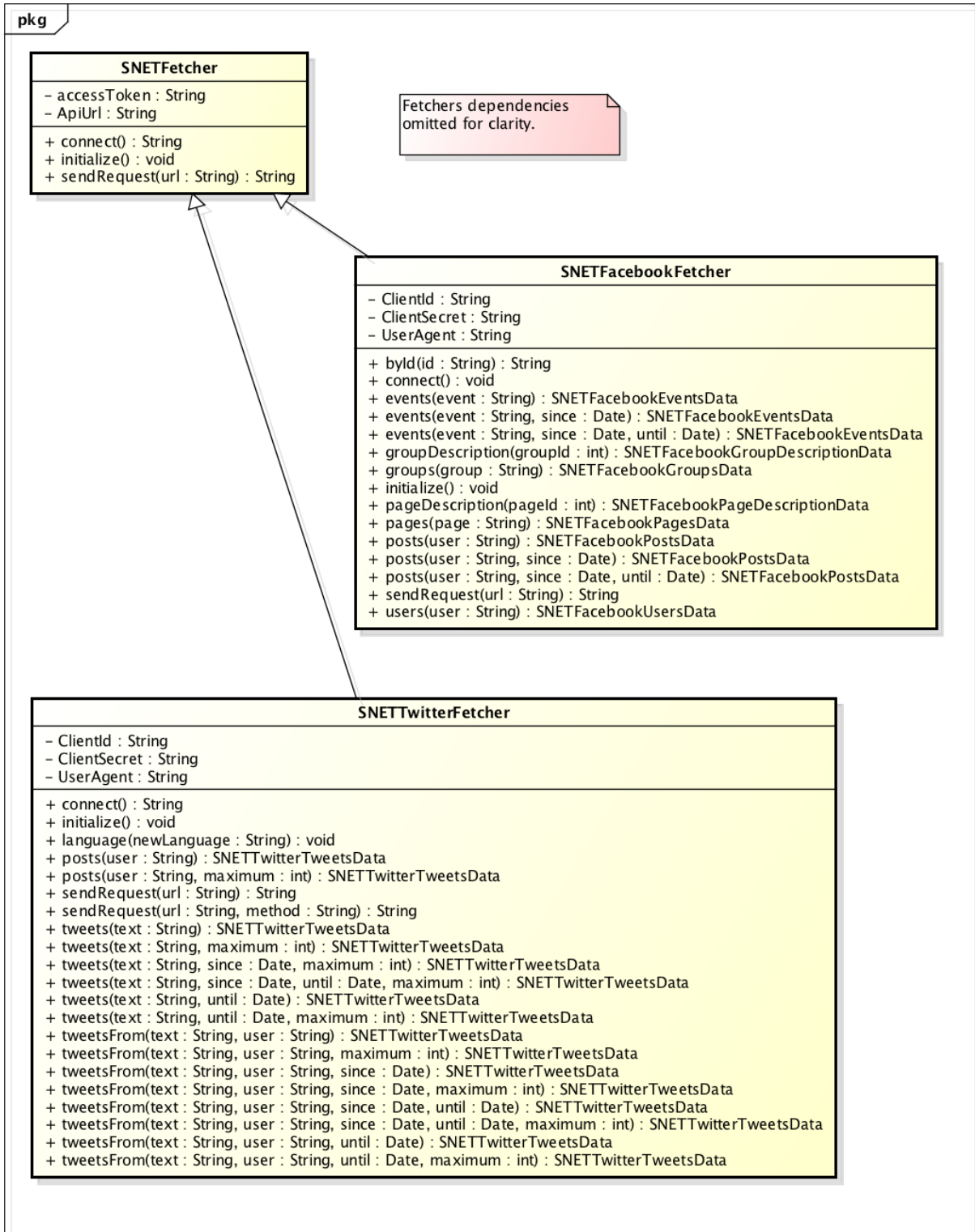
Then there are the GODData clases, these ones receive the JSON response from a request made by a fetcher and transforms it into a GODData type of object.

It's important to mention that both Facebook and Twitter APIs need a client id and secret to work, this can be done creating a specific type of user in the developer section of both APIs.

Last but not least, we have classes responsible to test all the methods within the fetchers and GODDdatas, keep in mind that we don't have control of the Facebook and Twitter servers, so some test may fail some times simply because the server response is not what was expected.

## 1.3 Class Diagram





## 1.4 Documentation

### 1.4.1 Fetchers

- Class SNETFetcher

SNETFetcher is a abstract class that contains the common parts used by SNETFacebookFetcher and SNETTwitterFetcher.

- Instance Variables

- \* `accessToken`

- Type:**

- String.

- Description:**

- This variable will store the access token necessary to access Facebook or Twitter servers.

- Class Variables

- \* `ApiUrl`

- Type:**

- String.

- Description:**

- This variable will store the full URL to either Facebook or Twitter servers.

- Functions

- \* `private`

- `sendRequest: url`

- Description:**

- This abstract function job is to send some HTTP request to some url, it will be specialized in SNETFacebookFetcher and SNETTwitterFetcher.

- \* `initialize-release`

- `initialize`

- Description:**

- This is the default function to initialize the class.

- \* `connection`

- `connection`

- Description:**

- This abstract function job is to connect to some server.

- Class SNETFacebookFetcher

SNETFacebookFetcher is a class that controls all the access to the Facebook API.

It sends requests to the facebook server and returns the result as a SNETFacebookUserData object, SNETFacebookPostsData, SNETFacebookPagesData, SNETFacebookPageDescriptionData, SNETFacebookGroupsData, SNETFacebookGroupDescriptionData and SNETFacebookEventsData.

– Class Variables

\* `ClientId`

**Type:**

String.

**Description:**

This variable will store the client id needed to connect to the Facebook API.

\* `ClientSecret`

**Type:**

String.

**Description:**

This variable will store the client secret needed to connect to the Facebook API.

\* `UserAgent`

**Type:**

String.

**Description:**

This variable will store the user agent used by `sendRequest` function to send HTTP requests to the Facebook API.

– Functions

\* `private`

· `sendRequest: url`

**Description:**

This function job is to send some HTTP request to Facebook API.

· `byId: id`

**Description:**

This function job is to be a helper to some functions that search in the Facebook API using some id.

\* `initialize-release`

· `initialize`

**Description:**

This is the default function to initialize the class, it will initialize all the class variables.

\* `connection`

· `connection`

**Description:**

This function job is to connect to Facebook API.

\* accessing

· **events:** event

**Description:**

This function job is to search for events with the parameter name.

· **events:** event **since:** since

**Description:**

This function job is to search for events with the parameter name since some date.

· **events:** event **since:** since **until:** until

**Description:**

This function job is to search for events with the parameter name since some date until another date.

· **groupDescription:** groupId

**Description:**

This function job is to get group information by id.

· **groups:** group

**Description:**

This function job is to search for groups with the parameter name.

· **pageDescription:** pageId

**Description:**

This function job is to get page information by id.

· **pages:** page

**Description:**

This function job is to search for pages with the parameter name.

· **posts:** user

**Description:**

This function job is to get posts from user.

· **posts:** user **since:** since

**Description:**

This function job is to get posts from user since some date.

```
posts: user since: since until: until
```

**Description:**

This function job is to get posts from user since some date until another date.

```
users: user
```

**Description:**

This function job is to search for users with the parameter name.

- Class SNETTwitterFetcher

SNETTwitterFetcher is a class that control all the access to the Twitter API.

It send requests to the twitter server and return the result as data class SNETTwitterTweetsData.

- Instance Variables

- \* `tokenType`

**Type:**

String.

**Description:**

This variable will store the token type necessary to perform different requests from the Twitter server.

- \* `lang`

**Type:**

String.

**Description:**

This variable will store the language used when searching for tweets.

- Class Variables

- \* `ClientId`

**Type:**

String.

**Description:**

This variable will store the client id needed to connect to the Twitter API.

- \* `ClientSecret`

**Type:**  
String.

**Description:**  
This variable will store the client secret needed to connect to the Twitter API.

\* **UserAgent**

**Type:**  
String.

**Description:**  
This variable will store the user agent used by `sendRequest` function to send HTTP requests to the Twitter API.

#### – Functions

\* **private**

· `sendRequest: url`

**Description:**  
This function job is to send some HTTP request to Twitter API.

· `sendRequest: url method: method`

**Description:**  
This function job is to send some HTTP request with either GET or POST method to Twitter API.

\* **initialize-release**

· `initialize`

**Description:**  
This is the default function to initialize the class, it will initialize all the class and instance variables.

\* **connection**

· `connection`

**Description:**  
This function job is to connect to Twitter API.

\* **accessing**

· `posts: user`

**Description:**  
This function job is to get posts from user.

· `posts: user maximum: max`



**Description:**

This function job is to get posts from user specifying the maximum amount of posts.

```
· tweets: text
```

**Description:**

This function job is to get tweets from text.

```
· tweets: text maximum: max
```

**Description:**

This function job is to get tweets from text specifying the maximum amount of tweets.

```
· tweets: text since: date maximum: max
```

**Description:**

This function job is to get tweets from text since date specifying the maximum amount of tweets.

```
· tweets: text since: initDate until: endDate maximum: max
```

**Description:**

This function job is to get tweets from text since date until another date specifying the maximum amount of tweets.

```
· tweets: text until: date
```

**Description:**

This function job is to get tweets from text until date.

```
· tweets: text until: date maximum: max
```

**Description:**

This function job is to get tweets from text until date specifying the maximum amount of tweets.

```
· tweetsFrom: text user: user
```

**Description:**

This function job is to get tweets from text from user.

```
· tweetsFrom: text user: user maximum: max
```

**Description:**

This function job is to get tweets from text from user specifying the maximum amount of tweets.

· `tweetsFrom: text user: user since: date`

**Description:**

This function job is to get tweets from text from user since date.

· `tweetsFrom: text user: user since: date maximum: max`

**Description:**

This function job is to get tweets from text from user since date specifying the maximum amount of tweets.

· `tweetsFrom: text user: user since: initDate until: endDate maximum: max`

**Description:**

This function job is to get tweets from text from user since date until another date specifying the maximum amount of tweets.

· `tweetsFrom: text user: user until: date`

**Description:**

This function job is to get tweets from text from user until date.

· `tweetsFrom: text user: user until: date maximum: max`

**Description:**

This function job is to get tweets from text from user until date specifying the maximum amount of tweets.

\* setting

· `language: newLanguage`

**Description:**

This function job is to set the language limiting the tweets searched by it.

### 1.4.2 GODDdatas

- Class SNETTwitterTweetsData

SNETTwitterTweetsData is a class that contains a list of SNETTwitterTweet.

– Instance Variables

\* `tweets`

**Type:**

Collection.

**Description:**

This variable will store a list of tweets.

– Functions

\* private  
· addTweet: json

**Description:**

This function job is to add a new tweet from a json to the collection.

\* initialize-release  
· initialize

**Description:**

This is the default function to initialize the class.

\* accessing  
· size

**Description:**

This function job is to show how much tweets the collection contains.

· tweet: position

**Description:**

This function job is to get a tweet at a position in the collection.

· tweets

**Description:**

This function job is to get the list of tweets.

- Class SNETTwitterTweet  
SNETTwitterTweet is a class that contains a Twitter tweet.

– Instance Variables

\* id

**Type:**

Integer.

**Description:**

This variable will store the id of the tweet.

\* message

**Type:**

String.

**Description:**

This variable will store the message of the tweet.

\* retweets

**Type:**

Integer.

**Description:**

This variable will store the number of retweets of the tweet.

\* favorited

**Type:**

Integer.

**Description:**

This variable will store the number of favorited of the tweet.

– Functions

\* private

· favorited: number

**Description:**

This function job is to set the number of favorited from tweet.

· id: idNumber

**Description:**

This function job is to set the id from tweet.

· message: text

**Description:**

This function job is to set the message from tweet.

· retweets: number

**Description:**

This function job is to set the number of retweets from tweet.

\* accessing

· favorited

**Description:**

This function job is to get the number of favorited from tweet.

· id

**Description:**

This function job is to get the id from tweet.

- `message`

**Description:**

This function job is to get the message from tweet.

- `retweets`

**Description:**

This function job is to get the number of retweets from tweet.

- Class `SNETFacebookPostsData`

`SNETFacebookPostsData` is a class that contains a list of `SNETFacebookPost`.

- Instance Variables

- \* `posts`

**Type:**

Collection.

**Description:**

This variable will store a list of posts.

- Functions

- \* `private`

- `addPost: json`

**Description:**

This function job is to add a new post from a json to the collection.

- \* `initialize-release`

- `initialize`

**Description:**

This is the default function to initialize the class.

- \* `accessing`

- `size`

**Description:**

This function job is to show how much posts the collection contains.

- `post: position`

**Description:**

This function job is to get a post at a position in the collection.

- `posts`

**Description:**

This function job is to get the list of posts.

- Class SNETFacebookPost

SNETFacebookPost is a class that contains a Facebook post.

- Instance Variables

- \* **message**

- Type:**

- String.

- Description:**

- This variable will store the message of the post.

- \* **type**

- Type:**

- String.

- Description:**

- This variable will store the type of the post.

- \* **caption**

- Type:**

- String.

- Description:**

- This variable will store the caption of the post.

- \* **description**

- Type:**

- String.

- Description:**

- This variable will store the description of the post.

- \* **picture**

- Type:**

- String.

- Description:**

- This variable will store the picture URL of the post.

\* `likes`

**Type:**

Integer.

**Description:**

This variable will store the number of likes of the post.

\* `shares`

**Type:**

Integer.

**Description:**

This variable will store the number of shares of the post.

\* `comments`

**Type:**

Collection.

**Description:**

This variable will store a list of comments of the post.

– Functions

\* `private`

· `addComment: json`

**Description:**

This function job is to add a new comment from a json to the collection.

· `caption: text`

**Description:**

This function job is to set the caption from post.

· `description: text`

**Description:**

This function job is to set the description from post.

· `likes: number`

**Description:**

This function job is to set the number of likes from post.

· `message: text`

**Description:**

This function job is to set the message from post.

- `picture: url`

**Description:**

This function job is to set the picture url from post.

- `shares: number`

**Description:**

This function job is to set the number of shares from post.

- `type: postType`

**Description:**

This function job is to set the type of post from post.

\* accessing

- `comment: position`

**Description:**

This function job is to get a comment from the collection.

- `caption`

**Description:**

This function job is to get the caption from post.

- `description`

**Description:**

This function job is to get the description from post.

- `likes`

**Description:**

This function job is to get the number of likes from post.

- `message`

**Description:**

This function job is to get the message from post.

- `picture`

**Description:**



This function job is to get the picture url from post.

- **shares**

**Description:**

This function job is to get the number of shares from post.

- **type**

**Description:**

This function job is to get the type of post from post.

- Class SNETFacebookComment

SNETFacebookComment is a class that contains a Facebook comment.

- Instance Variables

- \* **message**

- Type:**

- String.

- Description:**

- This variable will store the message of the comment.

- \* **id**

- Type:**

- Integer.

- Description:**

- This variable will store the id of the comment.

- \* **likes**

- Type:**

- Integer.

- Description:**

- This variable will store the number of likes of the comment.

- Functions

- \* **private**

- **id: idNumber**

- Description:**

- This function job is to set the id from comment.

- **likes: number**

**Description:**

This function job is to set the number of likes from comment.

- **message: text**

**Description:**

This function job is to set the message from comment.

- \* accessing

- **id**

**Description:**

This function job is to get the id from comment.

- **likes**

**Description:**

This function job is to get the number of likes from comment.

- **message**

**Description:**

This function job is to get the message from comment.

- **Class SNETFacebookPagesData**

SNETFacebookPagesData is a class that contains a list of SNETFacebookPage.

- Instance Variables

- \* **pages**

**Type:**

Collection.

**Description:**

This variable will store a list of pages.

- Functions

- \* private

- **addPage: json**

**Description:**

This function job is to add a new page from a json to the collection.

- \* initialize-release

- **initialize**

**Description:**

This is the default function to initialize the class.

\* accessing

· size

**Description:**

This function job is to show how much pages the collection contains.

· page: position

**Description:**

This function job is to get a page at a position in the collection.

· pages

**Description:**

This function job is to get the list of pages.

- Class SNETFacebookPage

SNETFacebookPage is a class that contains a Facebook page.

- Instance Variables

\* id

**Type:**

Integer.

**Description:**

This variable will store the id of the page.

\* category

**Type:**

String.

**Description:**

This variable will store the category of the page.

\* name

**Type:**

String.

**Description:**

This variable will store the name of the page.

- Functions

\* private

· category: pageCategory

**Description:**

This function job is to set the category from page.

· id: idNumber

**Description:**

This function job is to set the id from page.

· name: pageName

**Description:**

This function job is to set the name from page.

\* accessing

· category

**Description:**

This function job is to get the category from page.

· id

**Description:**

This function job is to get the id from page.

· name

**Description:**

This function job is to get the name from page.

- Class SNETFacebookPageDescriptionData

SNETFacebookPageDescriptionData is a class that contains the description of a certain Facebook page.

– Instance Variables

\* id

**Type:**

Integer.

**Description:**

This variable will store the id of the page.

\* category

**Type:**

String.

**Description:**

This variable will store the category of the page.

\* **name**

**Type:**

String.

**Description:**

This variable will store the name of the page.

\* **description**

**Type:**

String.

**Description:**

This variable will store the description of the page.

\* **likes**

**Type:**

Integer.

**Description:**

This variable will store the likes of the page.

– **Functions**

\* **private**

· **category:** pageCategory

**Description:**

This function job is to set the category of page.

· **description:** text

**Description:**

This function job is to set the description of page.

· **id:** idNumber

**Description:**

This function job is to set the id of page.

· **name:** pageName

**Description:**

This function job is to set the name of page.

- likes: number

**Description:**

This function job is to set the likes of page.

- \* accessing

- category

**Description:**

This function job is to get the category of page.

- description

**Description:**

This function job is to get the description of page.

- id

**Description:**

This function job is to get the id of page.

- name

**Description:**

This function job is to get the name of page

- likes

**Description:**

This function job is to get the likes of page.

- Class SNETFacebookGroupsData

SNETFacebookGroupsData is a class that contains a list of SNETFacebookGroup.

- Instance Variables

- \* groups

**Type:**

Collection.

**Description:**

This variable will store a list of groups.

- Functions

- \* private

- addGroup: json

**Description:**

This function job is to add a new group from a json to the collection.

- \* initialize-release

- **initialize**

**Description:**

This is the default function to initialize the class.

- \* accessing

- **size**

**Description:**

This function job is to show how much groups the collection contains.

- **group: position**

**Description:**

This function job is to get a group at a position in the collection.

- **groups**

**Description:**

This function job is to get the list of groups.

- Class SNETFacebookGroup

SNETFacebookGroup is a class that contains a Facebook group.

- Instance Variables

- \* **id**

**Type:**

Integer.

**Description:**

This variable will store the id of the group.

- \* **name**

**Type:**

String.

**Description:**

This variable will store the name of the group.

- Functions

- \* private

- **id: idNumber**

**Description:**

This function job is to set the id from group.

· `name: groupName`

**Description:**

This function job is to set the name from group.

\* accessing

· `id`

**Description:**

This function job is to get the id from group.

· `name`

**Description:**

This function job is to get the name from group.

- Class `SNETFacebookGroupDescriptionData`

`SNETFacebookGroupDescriptionData` is a class that contains the description of a certain Facebook group.

- Instance Variables

\* `id`

**Type:**

Integer.

**Description:**

This variable will store the id of the group.

\* `name`

**Type:**

String.

**Description:**

This variable will store the name of the group.

\* `description`

**Type:**

String.

**Description:**

This variable will store the description of the group.

- Functions

\* `private`



- `description: text`

**Description:**

This function job is to set the description of group.

- `id: idNumber`

**Description:**

This function job is to set the id of group.

- `name: groupName`

**Description:**

This function job is to set the name of group.

- \* `accessing`

- `description`

**Description:**

This function job is to get the description of group.

- `id`

**Description:**

This function job is to get the id of group.

- `name`

**Description:**

This function job is to get the name of group

- Class `SNETFacebookEventsData`  
`SNETFacebookEventsData` is a class that contains a list of `SNETFacebookEvent`.

- Instance Variables

- \* `events`

**Type:**

Collection.

**Description:**

This variable will store a list of events.

- Functions

- \* `private`

- `addEvent: json`

**Description:**

This function job is to add a new event from a json to the collection.

\* initialize-release

· **initialize**

**Description:**

This is the default function to initialize the class.

\* accessing

· **size**

**Description:**

This function job is to show how much events the collection contains.

· **event: position**

**Description:**

This function job is to get a event at a position in the collection.

· **events**

**Description:**

This function job is to get the list of events.

- Class SNETFacebookEvent

SNETFacebookEvent is a class that contains a Facebook event.

- Instance Variables

\* **id**

**Type:**

Integer.

**Description:**

This variable will store the id of the event.

\* **location**

**Type:**

String.

**Description:**

This variable will store the location of the event.

\* **name**

**Type:**

String.

**Description:**

This variable will store the name of the event.

– Functions

\* private

· location: eventLocation

**Description:**

This function job is to set the location from event.

· id: idNumber

**Description:**

This function job is to set the id from event.

· name: eventName

**Description:**

This function job is to set the name from event.

\* accessing

· location

**Description:**

This function job is to get the location from event.

· id

**Description:**

This function job is to get the id from event.

· name

**Description:**

This function job is to get the name from event.

• Class SNETFacebookUsersData

SNETFacebookUsersData is a class that contains a list of SNETFacebookUser.

– Instance Variables

\* users

**Type:**

Collection.

**Description:**

This variable will store a list of users.

– Functions

\* initialize-release

· **initialize**

**Description:**

This is the default function to initialize the class.

\* accessing

· **size**

**Description:**

This function job is to show how much users the collection contains.

· **user: position**

**Description:**

This function job is to get a user at a position in the collection.

· **users**

**Description:**

This function job is to get the list of users.

## 1.5 Contact

thd.araujo@gmail.com