

Jobify App-Server

Generated by Doxygen 1.8.7

Thu Dec 1 2016 15:48:26

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	Class Documentation	5
3.1	Database Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Constructor & Destructor Documentation	6
3.1.2.1	Database	6
3.1.3	Member Function Documentation	6
3.1.3.1	clear_all	6
3.1.3.2	close	6
3.1.3.3	config	6
3.1.3.4	erase	6
3.1.3.5	get	6
3.1.3.6	open	6
3.1.3.7	put	7
3.1.3.8	writeBatch	7
3.2	DatabaseLevelDB Class Reference	7
3.2.1	Constructor & Destructor Documentation	7
3.2.1.1	DatabaseLevelDB	7
3.2.2	Member Function Documentation	8
3.2.2.1	clear_all	8
3.2.2.2	close	8
3.2.2.3	config	8
3.2.2.4	erase	8
3.2.2.5	get	8
3.2.2.6	open	8
3.2.2.7	put	8
3.2.2.8	writeBatch	9

3.3	DatabaseMockRAM Class Reference	9
3.3.1	Detailed Description	9
3.3.2	Member Function Documentation	10
3.3.2.1	clear_all	10
3.3.2.2	config	10
3.3.2.3	erase	10
3.3.2.4	erase	10
3.3.2.5	get	10
3.3.2.6	open	10
3.3.2.7	put	10
3.3.2.8	put	10
3.3.2.9	writeBatch	11
3.4	DatabaseWriteBatch Class Reference	11
3.4.1	Detailed Description	11
3.4.2	Constructor & Destructor Documentation	12
3.4.2.1	DatabaseWriteBatch	12
3.4.3	Member Function Documentation	12
3.4.3.1	Erase	12
3.4.3.2	getKeys	12
3.4.3.3	getOperations	12
3.4.3.4	getValues	12
3.4.3.5	Put	12
3.4.4	Member Data Documentation	12
3.4.4.1	db	12
3.4.4.2	keys	12
3.4.4.3	operations	13
3.4.4.4	values	13
3.5	DBElement Class Reference	13
3.5.1	Detailed Description	14
3.5.2	Constructor & Destructor Documentation	14
3.5.2.1	DBElement	14
3.5.3	Member Function Documentation	14
3.5.3.1	_setKey	14
3.5.3.2	_setValue	14
3.5.3.3	_setValueVars	15
3.5.3.4	endBatch	15
3.5.3.5	erase	15
3.5.3.6	get	15
3.5.3.7	getKey	15
3.5.3.8	getKeyToString	15

3.5.3.9	getValue	15
3.5.3.10	getValueToString	15
3.5.3.11	put	15
3.5.3.12	reSync	16
3.5.3.13	setKey	16
3.5.3.14	setValue	16
3.5.3.15	startBatch	16
3.5.4	Member Data Documentation	16
3.5.4.1	batch	16
3.5.4.2	db	16
3.5.4.3	key	16
3.5.4.4	synced	16
3.5.4.5	value	16
3.6	DBElementMock Class Reference	17
3.6.1	Detailed Description	17
3.6.2	Constructor & Destructor Documentation	17
3.6.2.1	DBElementMock	17
3.7	DefaultHandler Class Reference	17
3.7.1	Detailed Description	18
3.7.2	Constructor & Destructor Documentation	18
3.7.2.1	DefaultHandler	18
3.8	DisLikeHandler Class Reference	18
3.8.1	Detailed Description	18
3.8.2	Constructor & Destructor Documentation	19
3.8.2.1	DisLikeHandler	19
3.9	EventHandler Class Reference	19
3.9.1	Detailed Description	19
3.9.2	Constructor & Destructor Documentation	19
3.9.2.1	EventHandler	19
3.9.3	Member Function Documentation	20
3.9.3.1	_handle	20
3.9.3.2	handle	20
3.9.4	Member Data Documentation	20
3.9.4.1	db	20
3.10	EventHandlerChecksAuthentication Class Reference	20
3.10.1	Detailed Description	20
3.10.2	Constructor & Destructor Documentation	21
3.10.2.1	EventHandlerChecksAuthentication	21
3.11	EventHandlerGrantsAuthentication Class Reference	21
3.11.1	Detailed Description	21

3.11.2	Constructor & Destructor Documentation	21
3.11.2.1	EventHandlerGrantsAuthentication	21
3.12	EventHandlerIgnoresAuthentication Class Reference	22
3.12.1	Detailed Description	22
3.12.2	Constructor & Destructor Documentation	22
3.12.2.1	EventHandlerIgnoresAuthentication	22
3.13	EventHandlerRemovesAuthentication Class Reference	22
3.13.1	Detailed Description	23
3.13.2	Constructor & Destructor Documentation	23
3.13.2.1	EventHandlerRemovesAuthentication	23
3.14	FriendDeleteHandler Class Reference	23
3.14.1	Detailed Description	23
3.14.2	Constructor & Destructor Documentation	23
3.14.2.1	FriendDeleteHandler	23
3.15	HandlerManager Class Reference	24
3.15.1	Detailed Description	24
3.15.2	Constructor & Destructor Documentation	24
3.15.2.1	HandlerManager	24
3.15.3	Member Function Documentation	25
3.15.3.1	handle	25
3.16	HerokuComHandler Class Reference	25
3.17	HttpRequest Class Reference	25
3.17.1	Detailed Description	27
3.18	HttpRequestConcrete Class Reference	27
3.18.1	Detailed Description	28
3.18.2	Constructor & Destructor Documentation	28
3.18.2.1	HttpRequestConcrete	28
3.18.3	Member Function Documentation	28
3.18.3.1	addValueToBody	28
3.18.3.2	getQueryCampo	28
3.18.3.3	getUri	28
3.18.3.4	getUriType	29
3.18.3.5	setResponse	29
3.19	HttpRequestMock Class Reference	29
3.19.1	Detailed Description	30
3.19.2	Constructor & Destructor Documentation	30
3.19.2.1	HttpRequestMock	30
3.19.3	Member Function Documentation	30
3.19.3.1	addValueToBody	30
3.19.3.2	getHeaderValue	31

3.20 JobGetHandler Class Reference	31
3.20.1 Detailed Description	31
3.20.2 Constructor & Destructor Documentation	31
3.20.2.1 JobGetHandler	31
3.21 JsonSerializer Class Reference	31
3.21.1 Detailed Description	32
3.21.2 Member Function Documentation	33
3.21.2.1 addArrayOfObjectToList	33
3.21.2.2 addBoolToList	33
3.21.2.3 addNumberToList	33
3.21.2.4 addValueToArrayList	33
3.21.2.5 addValueToObject	33
3.21.2.6 addValueToObjectList	33
3.21.2.7 get	33
3.21.2.8 get	33
3.21.2.9 joinValueIntoList	34
3.21.2.10 removeBegAndEndQuotes	34
3.21.2.11 turnArrayListToArray	34
3.21.2.12 turnArrayListToArray	34
3.21.2.13 turnMapToObj	34
3.21.2.14 turnObjectListToObject	34
3.21.2.15 turnObjectListToObject	34
3.21.2.16 turnPairsVectorToObject	34
3.21.2.17 turnVectorToArray	35
3.22 LikeHandler Class Reference	35
3.22.1 Detailed Description	35
3.22.2 Constructor & Destructor Documentation	35
3.22.2.1 LikeHandler	35
3.23 LogInHandler Class Reference	35
3.23.1 Detailed Description	36
3.23.2 Constructor & Destructor Documentation	36
3.23.2.1 LogInHandler	36
3.24 LogOutHandler Class Reference	36
3.24.1 Detailed Description	36
3.24.2 Constructor & Destructor Documentation	36
3.24.2.1 LogOutHandler	36
3.25 RespondFriendRequestHandler Class Reference	37
3.25.1 Detailed Description	37
3.25.2 Constructor & Destructor Documentation	37
3.25.2.1 RespondFriendRequestHandler	37

3.26	SendFriendRequestHandler Class Reference	37
3.26.1	Detailed Description	38
3.26.2	Constructor & Destructor Documentation	38
3.26.2.1	SendFriendRequestHandler	38
3.27	Server Class Reference	38
3.27.1	Detailed Description	38
3.27.2	Constructor & Destructor Documentation	38
3.27.2.1	Server	38
3.27.3	Member Function Documentation	39
3.27.3.1	createListeningConnection	39
3.27.3.2	poll	39
3.27.3.3	stop	39
3.28	Server_Logger Class Reference	39
3.28.1	Detailed Description	39
3.28.2	Member Function Documentation	39
3.28.2.1	getInstance	39
3.29	SignUpHandler Class Reference	40
3.29.1	Detailed Description	40
3.29.2	Constructor & Destructor Documentation	40
3.29.2.1	SignUpHandler	40
3.30	SkillGetHandler Class Reference	40
3.30.1	Detailed Description	41
3.30.2	Constructor & Destructor Documentation	41
3.30.2.1	SkillGetHandler	41
3.31	TokenAuthenticator Class Reference	41
3.31.1	Detailed Description	41
3.31.2	Member Function Documentation	41
3.31.2.1	createToken	41
3.31.2.2	isValidToken	42
3.31.2.3	removeToken	42
3.32	TopJobHandler Class Reference	42
3.32.1	Detailed Description	42
3.32.2	Constructor & Destructor Documentation	42
3.32.2.1	TopJobHandler	42
3.33	TopRankGetHandler Class Reference	43
3.33.1	Detailed Description	43
3.33.2	Constructor & Destructor Documentation	43
3.33.2.1	TopRankGetHandler	43
3.34	url_data Struct Reference	43
3.35	User Class Reference	43

3.35.1 Detailed Description	44
3.35.2 Constructor & Destructor Documentation	44
3.35.2.1 User	44
3.35.3 Member Function Documentation	44
3.35.3.1 _setKey	44
3.35.3.2 _setValue	45
3.35.3.3 _setValueVars	45
3.35.3.4 DBcreate	45
3.35.3.5 DBget	45
3.35.3.6 setPassword	45
3.35.3.7 setUsername	45
3.36 UserDeleteHandler Class Reference	45
3.36.1 Detailed Description	46
3.36.2 Constructor & Destructor Documentation	46
3.36.2.1 UserDeleteHandler	46
3.37 UserGetHandler Class Reference	46
3.37.1 Detailed Description	46
3.37.2 Constructor & Destructor Documentation	46
3.37.2.1 UserGetHandler	46
3.38 UserGetProfileHandler Class Reference	47
3.38.1 Detailed Description	47
3.38.2 Constructor & Destructor Documentation	47
3.38.2.1 UserGetProfileHandler	47
3.39 UserMetadata Class Reference	47
3.39.1 Detailed Description	49
3.39.2 Constructor & Destructor Documentation	50
3.39.2.1 UserMetadata	50
3.39.3 Member Function Documentation	50
3.39.3.1 _setKey	50
3.39.3.2 _setValue	50
3.39.3.3 _setValueVars	50
3.39.3.4 addFriend	50
3.39.3.5 addLike	50
3.39.3.6 addNotificacion	50
3.39.3.7 changeEmail	50
3.39.3.8 changeJobList	51
3.39.3.9 changeName	51
3.39.3.10 changePic	51
3.39.3.11 changeResume	51
3.39.3.12 changeSkillList	51

3.39.3.13 changeUltimaUbicacion	51
3.39.3.14 getProfile	51
3.39.3.15 removeFriend	51
3.39.3.16 removeLike	51
3.39.3.17 removeNotification	52
3.39.3.18 setJoinDate	52
3.39.3.19 setUsername	52
3.40 UserModifyHandler Class Reference	52
3.40.1 Detailed Description	52
3.40.2 Constructor & Destructor Documentation	52
3.40.2.1 UserModifyHandler	52
Index	54

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Database	5
DatabaseLevelDB	7
DatabaseMockRAM	9
DatabaseWriteBatch	11
DBElement	13
DBElementMock	17
User	43
UserMetadata	47
EventHandler	19
EventHandlerChecksAuthentication	20
FriendDeleteHandler	23
RespondFriendRequestHandler	37
SendFriendRequestHandler	37
UserGetHandler	46
UserModifyHandler	52
EventHandlerGrantsAuthentication	21
LoginHandler	35
EventHandlerIgnoresAuthentication	22
DefaultHandler	17
DisLikeHandler	18
JobGetHandler	31
LikeHandler	35
SignUpHandler	40
SkillGetHandler	40
TopJobHandler	42
TopRankGetHandler	43
UserGetProfileHandler	47
EventHandlerRemovesAuthentication	22
LogoutHandler	36
UserDeleteHandler	45
HandlerManager	24
HerokuComHandler	25
HttpRequest	25
HttpRequestConcrete	27
HttpRequestMock	29
JsonSerializer	31

Server	38
Server_Logger	39
TokenAuthenticator	41
url_data	43

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Database	
Clase abstracta como interface para base de datos	5
DatabaseLevelDB	7
DatabaseMockRAM	
Clase mock que maneja la base de datos	9
DatabaseWriteBatch	
Clase que maneja el batch	11
DBElement	
Clase que representa a los registros de la base de datos	13
DBElementMock	
Clase mock que representa a los registros de la base de datos	17
DefaultHandler	
Clase handler que maneja requests invalidos	17
DisLikeHandler	
Clase handler que elimina usuario	18
EventHandler	
Clase madre para todos los event handlers	19
EventHandlerChecksAuthentication	
Clase madre para algunos EventHandlers	20
EventHandlerGrantsAuthentication	
Clase madre para algunos EventHandlers	21
EventHandlerIgnoresAuthentication	
Clase madre para algunos EventHandlers	22
EventHandlerRemovesAuthentication	
Clase madre para algunos EventHandlers	22
FriendDeleteHandler	
Clase handler que elimina usuario	23
HandlerManager	
Clase que maneja los handlers	24
HerokuComHandler	25
HttpRequest	
Clase que se encarga de manejar httprequests	25
HttpRequestConcrete	
Clase que maneja las httprequests	27
HttpRequestMock	
Clase que maneja las httprequests	29
JobGetHandler	
Clase handler que obtiene un usuario	31

JsonSerializer	Clase que se encarga de tratar los archivos en formato json	31
LikeHandler	Clase handler que elimina usuario	35
LoginHandler	Clase handler que se encarga del login	35
LogoutHandler	Clase handler se encarga del logout	36
RespondFriendRequestHandler	Clase handler que responde notificacion de agregar amigo	37
SendFriendRequestHandler	Clase handler que envia notificacion de agregar amigo	37
Server	Clase que maneja el server	38
Server_Logger	Clase que maneja el logger	39
SignUpHandler	Clase handler que se encarga del signup	40
SkillGetHandler	Clase handler que obtiene un usuario	40
TokenAuthenticator	Clase que se encarga de manejar tokens	41
TopJobHandler	Clase handler que obtiene un usuario	42
TopRankGetHandler	Clase handler que obtiene un usuario	43
url_data		43
User	Clase que maneja al usuario	43
UserDeleteHandler	Clase handler que elimina usuario	45
UserGetHandler	Clase handler que obtiene un usuario	46
UserGetProfileHandler	Clase que devuelve el perfil de otros usuarios	47
UserMetadata	Clase que contiene la informacion del usuario	47
UserModifyHandler	Clase handler que modifica un usuario	52

Chapter 3

Class Documentation

3.1 Database Class Reference

Clase abstracta como interface para base de datos.

```
#include <Database.h>
```

Inheritance diagram for Database:

Public Member Functions

- [Database](#) ()
Funcion que inicializa la clase.
- virtual Status [config](#) (const std::string &db_path, bool, bool=false)=0
Funcion que establece el path a la base de datos.
- virtual Status [open](#) ()=0
Funcion que abre la base de datos para ser leida o modificada.
- virtual void [close](#) ()=0
Funcion que cierra la base de datos.
- virtual Status [clear_all](#) ()=0
Funcion que elimina todos los elementos de la base de datos.
- virtual Status [erase](#) ([DBElement](#) &elem)=0
Funcion que elimina un elemento de la base de datos.
- virtual Status [put](#) ([DBElement](#) &elem)=0
Funcion que agrega un elemento a la base de datos.
- virtual Status [get](#) ([DBElement](#) &elem)=0
Funcion que obtiene un elemento de la base de datos.
- virtual Status [writeBatch](#) ([DatabaseWriteBatch](#) *batch)=0
Funcion que procesa el batch.

3.1.1 Detailed Description

Clase abstracta como interface para base de datos.

Esta clase sirve como interface para las clases \ [DatabaseLevelDB](#) y [DatabaseMockRAM](#).

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Database::Database () [inline]

Funcion que inicializa la clase.

Esta funcion es redefinida en las clases hijas.

3.1.3 Member Function Documentation

3.1.3.1 virtual Status Database::clear_all () [pure virtual]

Funcion que elimina todos los elementos de la base de datos.

Esta funcion es redefinida en las clases hijas.

Implemented in [DatabaseLevelDB](#), and [DatabaseMockRAM](#).

3.1.3.2 virtual void Database::close () [pure virtual]

Funcion que cierra la base de datos.

Esta funcion es redefinida en las clases hijas.

Implemented in [DatabaseLevelDB](#), and [DatabaseMockRAM](#).

3.1.3.3 virtual Status Database::config (const std::string & db_path, bool , bool = false) [pure virtual]

Funcion que establece el path a la base de datos.

Esta funcion es redefinida en las clases hijas.

Implemented in [DatabaseLevelDB](#), and [DatabaseMockRAM](#).

3.1.3.4 virtual Status Database::erase (DBElement & elem) [pure virtual]

Funcion que elimina un elemento de la base de datos.

Esta funcion es redefinida en las clases hijas.

Implemented in [DatabaseLevelDB](#), and [DatabaseMockRAM](#).

3.1.3.5 virtual Status Database::get (DBElement & elem) [pure virtual]

Funcion que obtiene un elemento de la base de datos.

Esta funcion es redefinida en las clases hijas.

Implemented in [DatabaseLevelDB](#), and [DatabaseMockRAM](#).

3.1.3.6 virtual Status Database::open () [pure virtual]

Funcion que abre la base de datos para ser leida o modificada.

Esta funcion es redefinida en las clases hijas.

Implemented in [DatabaseLevelDB](#), and [DatabaseMockRAM](#).

3.1.3.7 virtual Status Database::put (DBElement & *elem*) [pure virtual]

Funcion que agrega un elemento a la base de datos.

Esta funcion es redefinida en las clases hijas.

Implemented in [DatabaseLevelDB](#), and [DatabaseMockRAM](#).

3.1.3.8 virtual Status Database::writeBatch (DatabaseWriteBatch * *batch*) [pure virtual]

Funcion que procesa el batch.

Esta funcion es redefinida en las clases hijas.

Implemented in [DatabaseLevelDB](#), and [DatabaseMockRAM](#).

The documentation for this class was generated from the following file:

- Database.h

3.2 DatabaseLevelDB Class Reference

Inheritance diagram for DatabaseLevelDB:

Collaboration diagram for DatabaseLevelDB:

Public Member Functions

- [DatabaseLevelDB](#) ()
Funcion que inicializa la clase.
- Status [config](#) (const std::string &db_path, bool create_if_missing, bool clean=false)
Funcion que establece el path a la base de datos.
- Status [open](#) ()
Funcion que abre la base de datos para ser leida o modificada.
- void [close](#) ()
Funcion que cierra la base de datos.
- Status [clear_all](#) ()
Funcion que elimina todos los elementos de la base de datos.
- Status [erase](#) (DBElement &elem)
Funcion que elimina un elemento de la base de datos.
- Status [put](#) (DBElement &elem)
Funcion que agrega un elemento a la base de datos.
- Status [get](#) (DBElement &elem)
Funcion que obtiene un elemento de la base de datos.
- Status [writeBatch](#) (DatabaseWriteBatch *batch)
Funcion que procesa el batch.

3.2.1 Constructor & Destructor Documentation

3.2.1.1 DatabaseLevelDB::DatabaseLevelDB ()

Funcion que inicializa la clase.

Inicializa db en NULL y deb_path en "".

3.2.2 Member Function Documentation

3.2.2.1 Status DatabaseLevelDB::clear_all () [virtual]

Funcion que elimina todos los elementos de la base de datos.

Si db no es NULL elimina los elementos de la base de datos.

Implements [Database](#).

3.2.2.2 void DatabaseLevelDB::close () [virtual]

Funcion que cierra la base de datos.

Si db no es NULL, lo convierte en NULL.

Implements [Database](#).

3.2.2.3 Status DatabaseLevelDB::config (const std::string & db_path, bool create_if_missing, bool clean = false) [virtual]

Funcion que establece el path a la base de datos.

Recibe un string con el path donde debe establecer la base de datos\ Si ya estaba abierta no puede ser reconfigurada\ Tambien recibe un bool y lo asigna a create_if_missing.

Implements [Database](#).

3.2.2.4 Status DatabaseLevelDB::erase (DBElement & elem) [virtual]

Funcion que elimina un elemento de la base de datos.

Recibe un elemento, lo busca a partir de su key y lo elimina.

Implements [Database](#).

3.2.2.5 Status DatabaseLevelDB::get (DBElement & elem) [virtual]

Funcion que obtiene un elemento de la base de datos.

Recibe un elemento, lo busca a partir de su key y lo obtiene.

Implements [Database](#).

3.2.2.6 Status DatabaseLevelDB::open () [virtual]

Funcion que abre la base de datos para ser leida o modificada.

Si el db no es NULL o si el db_path es "" devuelve error\ Si no ubica el db con el db_path.

Implements [Database](#).

3.2.2.7 Status DatabaseLevelDB::put (DBElement & elem) [virtual]

Funcion que agrega un elemento a la base de datos.

Recibe un elemento y lo introduce a la base de datos.

Implements [Database](#).

3.2.2.8 Status DatabaseLevelDB::writeBatch (DatabaseWriteBatch * batch) [virtual]

Funcion que procesa el batch.

Recibe un Databasewritebatch y lo procesa para aplicarlo en la base de datos.

Implements [Database](#).

The documentation for this class was generated from the following files:

- DatabaseLevelDB.h
- DatabaseLevelDB.cpp

3.3 DatabaseMockRAM Class Reference

Clase mock que maneja la base de datos.

```
#include <DatabaseMockRAM.h>
```

Inheritance diagram for DatabaseMockRAM:

Collaboration diagram for DatabaseMockRAM:

Public Member Functions

- [DatabaseMockRAM](#) ()
Funcion que inicializa la clase.
- Status [config](#) (const std::string &db_path, bool, bool)
Funcion que establece el path a la base de datos.
- Status [open](#) ()
Funcion que abre la base de datos para ser leida o modificada.
- void [close](#) ()
Funcion que cierra la base de datos.
- Status [clear_all](#) ()
Funcion que elimina todos los elementos de la base de datos.
- Status [erase](#) (DBElement &elem)
Funcion que elimina un elemento de la base de datos.
- Status [erase](#) (std::string key)
Funcion que elimina un elemento de la base de datos.
- Status [put](#) (DBElement &elem)
Funcion que agrega un elemento a la base de datos.
- Status [put](#) (std::string key, std::string value)
Funcion que agrega un elemento a la base de datos.
- Status [get](#) (DBElement &elem)
Funcion que obtiene un elemento de la base de datos.
- Status [writeBatch](#) (DatabaseWriteBatch *batch)
Funcion que procesa el batch.

3.3.1 Detailed Description

Clase mock que maneja la base de datos.

Esta clase emplea como interface [Database](#) y sirve de mock para subir a travis\ Utiliza para esto la clase map de C++

3.3.2 Member Function Documentation

3.3.2.1 Status DatabaseMockRAM::clear_all () [virtual]

Funcion que elimina todos los elementos de la base de datos.

Utiliza la funcion clear de la clase map.

Implements [Database](#).

3.3.2.2 Status DatabaseMockRAM::config (const std::string & db_path, bool create_if_missing, bool clean) [virtual]

Funcion que establece el path a la base de datos.

Devuelve status OK.

Implements [Database](#).

3.3.2.3 Status DatabaseMockRAM::erase (DBElement & elem) [virtual]

Funcion que elimina un elemento de la base de datos.

Recibe un elemento, lo busca a partir de su key y lo elimina.

Implements [Database](#).

3.3.2.4 Status DatabaseMockRAM::erase (std::string key)

Funcion que elimina un elemento de la base de datos.

Recibe una key de un elemento, lo busca y lo elimina.

3.3.2.5 Status DatabaseMockRAM::get (DBElement & elem) [virtual]

Funcion que obtiene un elemento de la base de datos.

Recibe un elemento, lo busca a partir de su key y lo obtiene.

Implements [Database](#).

3.3.2.6 Status DatabaseMockRAM::open () [virtual]

Funcion que abre la base de datos para ser leida o modificada.

Devuelve status OK.

Implements [Database](#).

3.3.2.7 Status DatabaseMockRAM::put (DBElement & elem) [virtual]

Funcion que agrega un elemento a la base de datos.

Recibe un elemento y lo introduce a la base de datos.

Implements [Database](#).

3.3.2.8 Status DatabaseMockRAM::put (std::string key, std::string value)

Funcion que agrega un elemento a la base de datos.

Recibe una key de un elemento, lo busca y lo agrega junto con su value.

3.3.2.9 Status DatabaseMockRAM::writeBatch (DatabaseWriteBatch * batch) [virtual]

Funcion que procesa el batch.

Recibe un Databasewritebatch y lo procesa para aplicarlo en la base de datos.

Implements [Database](#).

The documentation for this class was generated from the following files:

- DatabaseMockRAM.h
- DatabaseMockRAM.cpp

3.4 DatabaseWriteBatch Class Reference

Clase que maneja el batch.

```
#include <DatabaseWriteBatch.h>
```

Collaboration diagram for DatabaseWriteBatch:

Public Member Functions

- [DatabaseWriteBatch](#) ([Database](#) * _db)
Funcion que inicializa la clase.
- void [Erase](#) ([DBElement](#) *elem)
Funcion que guarda la accion de eliminar un elemento.
- void [Put](#) ([DBElement](#) *elem)
Funcion que guarda la accion de agregar un elemento.
- std::vector< std::string > * [getKeys](#) ()
Funcion que devuelve el vector keys.
- std::vector< std::string > * [getValues](#) ()
Funcion que devuelve el vector values.
- std::vector< std::string > * [getOperations](#) ()
Funcion que devuelve el vector operations.

Protected Attributes

- [Database](#) * db
Variable puntero db.
- std::vector< std::string > [keys](#)
Variable vector keys.
- std::vector< std::string > [values](#)
Variable vector values.
- std::vector< std::string > [operations](#)
Variable vector operations.

3.4.1 Detailed Description

Clase que maneja el batch.

Se encarga de administrar las funciones que se pueden colocar en el batch.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 DatabaseWriteBatch::DatabaseWriteBatch (Database * *_db*)

Funcion que inicializa la clase.

Recibe un puntero a una clase [Database](#) y lo guarda en su variable db.

3.4.3 Member Function Documentation

3.4.3.1 void DatabaseWriteBatch::Erase (DBElement * *elem*)

Funcion que guarda la accion de eliminar un elemento.

Recibe un elemento y lo agrega a las variables keys y values, asi como agrega erase a operations.

3.4.3.2 std::vector<std::string>* DatabaseWriteBatch::getKeys () [inline]

Funcion que devuelve el vector keys.

Devuelve el vector con las keys de los elementos que va a tratar el batch.

3.4.3.3 std::vector<std::string>* DatabaseWriteBatch::getOperations () [inline]

Funcion que devuelve el vector operations.

Devuelve el vector que tiene las acciones a tomar con los elementos almacenados en las otras variables.

3.4.3.4 std::vector<std::string>* DatabaseWriteBatch::getValues () [inline]

Funcion que devuelve el vector values.

Devuelve el vector con los valores de los elementos que va a tratar el batch.

3.4.3.5 void DatabaseWriteBatch::Put (DBElement * *elem*)

Funcion que guarda la accion de agregar un elemento.

Recibe un elemento y lo agrega a las variables keys y values, asi como agrega put a operations.

3.4.4 Member Data Documentation

3.4.4.1 Database* DatabaseWriteBatch::db [protected]

Variable puntero db.

Apunta a una [Database](#).

3.4.4.2 std::vector<std::string> DatabaseWriteBatch::keys [protected]

Variable vector keys.

Contiene las keys de los elementos que procesara el batch.

3.4.4.3 `std::vector<std::string> DatabaseWriteBatch::operations` [protected]

Variable vector operations.

Contiene las operaciones que procesara el batch.

3.4.4.4 `std::vector<std::string> DatabaseWriteBatch::values` [protected]

Variable vector values.

Contiene los valores de los elementos que procesara el batch.

The documentation for this class was generated from the following files:

- DatabaseWriteBatch.h
- DatabaseWriteBatch.cpp

3.5 DBElement Class Reference

Clase que representa a los registros de la base de datos.

```
#include <DBElement.h>
```

Inheritance diagram for DBElement:

Collaboration diagram for DBElement:

Public Member Functions

- [DBElement](#) ([Database](#) *db, [DatabaseWriteBatch](#) *dbbatch=NULL)
Funcion que inicializa DBElement.
- [std::string getKey](#) ()
Funcion que devuelve el valor key.
- [std::string getValue](#) ()
Funcion que devuelve el valor value.
- [std::string getKeyToString](#) ()
Funcion que devuelve el valor key.
- [std::string getValueToString](#) ()
Funcion que devuelve el valor vaue.
- void [setKey](#) (std::string key)
Funcion que establece la key.
- void [setValue](#) (std::string value)
Funcion que establece el value.
- Status [get](#) ()
Funcion que devuelve su estado en la base de datos.
- Status [put](#) ()
Funcion que agrega al dbelement a la base de datos.
- Status [erase](#) ()
Funcion que elimina el dbelement de la base de datos.
- void [startBatch](#) ()
- Status [endBatch](#) ()
Funcion que procesa el batch y lo elimina.
- void [reSync](#) ()
Funcion que fuerza la variable synced a false.

Protected Member Functions

- virtual void [_setKey](#) ()=0
Funcion que establece la key.
- virtual void [_setValue](#) ()=0
Funcion que establece el value.
- virtual void [_setValueVars](#) ()=0
Funcion que establece varias variables.

Protected Attributes

- std::string [key](#)
Variable key.
- std::string [value](#)
Variable value.
- [Database](#) * [db](#)
Variable db.
- [DatabaseWriteBatch](#) * [batch](#)
Variable batch.
- bool [synced](#)
Variable bool synced.

3.5.1 Detailed Description

Clase que representa a los registros de la base de datos.

Esta clase se encarga de obtener elementos o modificarlos en la base de datos usando leveldb\ De esta clase heredan luego los distintos tipos de registros que pueden darse.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 `DBElement::DBElement (Database * db, DatabaseWriteBatch * dbbatch = NULL)`

Funcion que inicializa [DBElement](#).

Se inicializa los valores de key y value con "" y db con el argumento que se pasa.

3.5.3 Member Function Documentation

3.5.3.1 `virtual void DBElement::_setKey () [protected],[pure virtual]`

Funcion que establece la key.

Las clases hijas generan a partir de sus variables la key para DBElements.

Implemented in [UserMetadata](#), and [User](#).

3.5.3.2 `virtual void DBElement::_setValue () [protected],[pure virtual]`

Funcion que establece el value.

Las clases hijas generan a partir de sus variables el value para DBElements.

Implemented in [UserMetadata](#), and [User](#).

3.5.3.3 `virtual void DBElement::_setValueVars () [protected],[pure virtual]`

Funcion que establece varias variables.

Las clases hijas procesan el valor value y a partir de esto establecen sus propias variables.

Implemented in [UserMetadata](#), and [User](#).

3.5.3.4 `Status DBElement::endBatch ()`

Funcion que procesa el batch y lo elimina.

Procesa el batch y luego lo elimina, asignando NULL a la variable batch.

3.5.3.5 `Status DBElement::erase ()`

Funcion que elimina el dbelement de la base de datos.

Agrega al bash la accion de eliminar el elemento para que luego cuando se ejecute se elimine de la base de datos.

3.5.3.6 `Status DBElement::get ()`

Funcion que devuelve su estado en la base de datos.

Se busca en la base de datos y devuelve si se encontro o no.

3.5.3.7 `std::string DBElement::getKey ()`

Funcion que devuelve el valor key.

Esta funcion se emplea para poder obtener la key de las clases hijas sin perder la abstraccion.

3.5.3.8 `std::string DBElement::getKeyToString () [inline]`

Funcion que devuelve el valor key.

Simplemente devuelve la key sin hacer ningun cambio.

3.5.3.9 `std::string DBElement::getValue ()`

Funcion que devuelve el valor value.

Esta funcion se emplea para poder obtener el value de las clases hijas sin perder la abstraccion.

3.5.3.10 `std::string DBElement::getValueToString () [inline]`

Funcion que devuelve el valor vaue.

Simplemente devuelve el valor value sin hacer ningun cambio.

3.5.3.11 `Status DBElement::put ()`

Funcion que agrega al dbelement a la base de datos.

Agrega al bash la accion de agregar el elemento para que luego cuando se ejecute se agregue a la base de datos.

3.5.3.12 void DBElement::reSync ()

Funcion que fuerza la variable synced a false.

Se emplea cuando se quiere forzar un get a pesar de que synced sea true.

3.5.3.13 void DBElement::setKey (std::string key) [inline]

Funcion que establece la key.

La funcion recibe un string y lo establece como key.

3.5.3.14 void DBElement::setValue (std::string value)

Funcion que establece el value.

Recibe un string de la base de datos y lo guarda en el valor value\ Ademias de esto emplea _setValueVars para que cada clase hija maneje estos datos como deba.

3.5.3.15 void DBElement::startBatch ()

Funcion que inicializa el batch. Con el puntero db crea un batch y lo asigna a la variable batch.

3.5.4 Member Data Documentation

3.5.4.1 DatabaseWriteBatch* DBElement::batch [protected]

Variable batch.

Es el puntero al batch.

3.5.4.2 Database* DBElement::db [protected]

Variable db.

Es el puntero a la base de datos.

3.5.4.3 std::string DBElement::key [protected]

Variable key.

Esta variable se emplea para interactuar con la base de datos.

3.5.4.4 bool DBElement::synced [protected]

Variable bool synced.

se fija si ya se hizo get de la base de datos

3.5.4.5 std::string DBElement::value [protected]

Variable value.

Contiene los valores que se encuentran en la base de datos en un string con formato json.

The documentation for this class was generated from the following files:

- DBElement.h
- DBElement.cpp

3.6 DBElementMock Class Reference

Clase mock que representa a los registros de la base de datos.

```
#include <DBElementMock.h>
```

Inheritance diagram for DBElementMock:

Collaboration diagram for DBElementMock:

Public Member Functions

- [DBElementMock](#) ([Database](#) *db, std::string c="", std::string v="", [DatabaseWriteBatch](#) *dbbatch=NULL)
Funcion que inicializa DBElement.
- void [setMockClave](#) (std::string s)
Funcion que establece mockclave.
- void [setMockValue](#) (std::string s)
Funcion que establece mockvalue.
- std::string [getMockClave](#) ()
Funcion que devuelve mockclave.
- std::string [getMockValue](#) ()
Funcion que devuelve mockvalue.

Additional Inherited Members

3.6.1 Detailed Description

Clase mock que representa a los registros de la base de datos.

Esta clase se encarga de obtener elementos o modificarlos en la base de datos.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 [DBElementMock::DBElementMock](#) ([Database](#) * db, std::string c = " ", std::string v = " ", [DatabaseWriteBatch](#) * dbbatch = NULL)

Funcion que inicializa [DBElement](#).

Se inicializa los valores de mock_clave y mock_value con "" ,dbbatch con NULL y db con el argumento que se pasa.

The documentation for this class was generated from the following files:

- DBElementMock.h
- DBElementMock.cpp

3.7 DefaultHandler Class Reference

Clase handler que maneja requests invalidos.

```
#include <DefaultHandler.h>
```

Inheritance diagram for DefaultHandler:

Collaboration diagram for DefaultHandler:

Public Member Functions

- [DefaultHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)

Funcion que inicializa la clase.

Additional Inherited Members

3.7.1 Detailed Description

Clase handler que maneja requests invalidos.

Hereda de [EventHandlerIgnoresAuthentication](#).

3.7.2 Constructor & Destructor Documentation

3.7.2.1 DefaultHandler::DefaultHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- DefaultHandler.h
- DefaultHandler.cpp

3.8 DisLikeHandler Class Reference

Clase handler que elimina usuario.

```
#include <DisLikeHandler.h>
```

Inheritance diagram for DisLikeHandler:

Collaboration diagram for DisLikeHandler:

Public Member Functions

- [DisLikeHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)

Funcion que inicializa la clase.

Additional Inherited Members

3.8.1 Detailed Description

Clase handler que elimina usuario.

Hereda de [EventHandlerIgnoresAuthentication](#).

3.8.2 Constructor & Destructor Documentation

3.8.2.1 DisLikeHandler::DisLikeHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- DisLikeHandler.h
- DisLikeHandler.cpp

3.9 EventHandler Class Reference

Clase madre para todos los event handlers.

```
#include <EventHandler.h>
```

Inheritance diagram for EventHandler:

Collaboration diagram for EventHandler:

Public Member Functions

- [EventHandler](#) ([Database](#) *db)
Funcion que inicializa [EventHandler](#).
- bool [handle](#) ([HttpRequest](#) &hmsg)
Funcion que maneja httprequests.

Protected Member Functions

- virtual void [_handle](#) ([HttpRequest](#) &hmsg)=0
Funcion que maneja httprequests.
- virtual void [_check_permission](#) ([HttpRequest](#) &hmsg)=0
- virtual bool [isMyRequest](#) ([HttpRequest](#) &hmsg)=0

Protected Attributes

- [Database](#) * db
Variable puntero db.

3.9.1 Detailed Description

Clase madre para todos los event handlers.

Contiene una funcion incializadora y luego dos funciones mas que son redefinidas por cada clase hija.

3.9.2 Constructor & Destructor Documentation

3.9.2.1 EventHandler::EventHandler (Database * db)

Funcion que inicializa [EventHandler](#).

Recibe como parametro un puntero a [Database](#) y lo almacena en su variable db.

3.9.3 Member Function Documentation

3.9.3.1 `virtual void EventHandler::_handle (HttpRequest & hmsg)` `[protected]`, `[pure virtual]`

Funcion que maneja httprequests.

Es redefinida en cada clase hija.

3.9.3.2 `bool EventHandler::handle (HttpRequest & hmsg)`

Funcion que maneja httprequests.

Es redefinida en cada clase hija.

3.9.4 Member Data Documentation

3.9.4.1 `Database* EventHandler::db` `[protected]`

Variable puntero db.

Contiene un puntero a una base de datos.

The documentation for this class was generated from the following files:

- `EventHandler.h`
- `EventHandler.cpp`

3.10 EventHandlerChecksAuthentication Class Reference

Clase madre para algunos EventHandlers.

```
#include <EventHandlerChecksAuthentication.h>
```

Inheritance diagram for EventHandlerChecksAuthentication:

Collaboration diagram for EventHandlerChecksAuthentication:

Public Member Functions

- [EventHandlerChecksAuthentication](#) (`Database *db`, `TokenAuthenticator *a`)

Funcion que inicializa la clase.

Protected Attributes

- [TokenAuthenticator](#) * `auth`

Variable puntero a un tokenauthenticator.

Additional Inherited Members

3.10.1 Detailed Description

Clase madre para algunos EventHandlers.

Clase de la que heredan eventhandlers que verifican autentificacion.

3.10.2 Constructor & Destructor Documentation

3.10.2.1 EventHandlerChecksAuthentication::EventHandlerChecksAuthentication (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- EventHandlerChecksAuthentication.h
- EventHandlerChecksAuthentication.cpp

3.11 EventHandlerGrantsAuthentication Class Reference

Clase madre para algunos EventHandlers.

```
#include <EventHandlerGrantsAuthentication.h>
```

Inheritance diagram for EventHandlerGrantsAuthentication:

Collaboration diagram for EventHandlerGrantsAuthentication:

Public Member Functions

- [EventHandlerGrantsAuthentication](#) (Database *db, TokenAuthenticator *a)
Funcion que inicializa la clase.

Protected Attributes

- [TokenAuthenticator](#) * auth
Variable puntero a un tokenauthenticator.

Additional Inherited Members

3.11.1 Detailed Description

Clase madre para algunos EventHandlers.

Clase de la que heredan eventhandlers que otorgan autentificacion.

3.11.2 Constructor & Destructor Documentation

3.11.2.1 EventHandlerGrantsAuthentication::EventHandlerGrantsAuthentication (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- EventHandlerGrantsAuthentication.h
- EventHandlerGrantsAuthentication.cpp

3.12 EventHandlerIgnoresAuthentication Class Reference

Clase madre para algunos EventHandlers.

```
#include <EventHandlerIgnoresAuthentication.h>
```

Inheritance diagram for EventHandlerIgnoresAuthentication:

Collaboration diagram for EventHandlerIgnoresAuthentication:

Public Member Functions

- [EventHandlerIgnoresAuthentication](#) ([Database](#) *db, [TokenAuthenticator](#) *a)
Funcion que inicializa la clase.

Protected Attributes

- [TokenAuthenticator](#) * auth
Variable puntero a un tokenauthenticator.

Additional Inherited Members

3.12.1 Detailed Description

Clase madre para algunos EventHandlers.

Clase de la que heredan eventhandlers que ignoran autentificacion.

3.12.2 Constructor & Destructor Documentation

3.12.2.1 EventHandlerIgnoresAuthentication::EventHandlerIgnoresAuthentication ([Database](#) * db, [TokenAuthenticator](#) * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- EventHandlerIgnoresAuthentication.h
- EventHandlerIgnoresAuthentication.cpp

3.13 EventHandlerRemovesAuthentication Class Reference

Clase madre para algunos EventHandlers.

```
#include <EventHandlerRemovesAuthentication.h>
```

Inheritance diagram for EventHandlerRemovesAuthentication:

Collaboration diagram for EventHandlerRemovesAuthentication:

Public Member Functions

- [EventHandlerRemovesAuthentication](#) ([Database](#) *db, [TokenAuthenticator](#) *a)
Funcion que inicializa la clase.

Protected Attributes

- [TokenAuthenticator](#) * [auth](#)

Variable puntero a un tokenauthenticator.

Additional Inherited Members

3.13.1 Detailed Description

Clase madre para algunos EventHandlers.

Clase de la que heredan eventhandlers que remueven autentificacion.

3.13.2 Constructor & Destructor Documentation

3.13.2.1 EventHandlerRemovesAuthentication::EventHandlerRemovesAuthentication ([Database](#) * *db*, [TokenAuthenticator](#) * *a*)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- [EventHandlerRemovesAuthentication.h](#)
- [EventHandlerRemovesAuthentication.cpp](#)

3.14 FriendDeleteHandler Class Reference

Clase handler que elimina usuario.

```
#include <FriendDeleteHandler.h>
```

Inheritance diagram for FriendDeleteHandler:

Collaboration diagram for FriendDeleteHandler:

Public Member Functions

- [FriendDeleteHandler](#) ([Database](#) **db*, [TokenAuthenticator](#) **a*)

Funcion que inicializa la clase.

Additional Inherited Members

3.14.1 Detailed Description

Clase handler que elimina usuario.

Hereda de [EventHandlerChecksAuthentication](#).

3.14.2 Constructor & Destructor Documentation

3.14.2.1 FriendDeleteHandler::FriendDeleteHandler ([Database](#) * *db*, [TokenAuthenticator](#) * *a*)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- FriendDeleteHandler.h
- FriendDeleteHandler.cpp

3.15 HandlerManager Class Reference

Clase que maneja los handlers.

```
#include <HandlerManager.h>
```

Collaboration diagram for HandlerManager:

Public Types

- enum **HandlerType** {
HANDLER_SIGNUP, **HANDLER_LOGIN**, **HANDLER_LOGOUT**, **HANDLER_GET_USER**,
HANDLER_DELETE_USER, **HANDLER_MODIFY_USER**, **HANDLER_GET_PROFILE_USER**, **HANDLER_SEND_FRIEND_REQUEST**,
HANDLER_RESPOND_FRIEND_REQUEST, **HANDLER_DEFAULT**, **HANDLER_TYPE_SIZE** }
- typedef enum
HandlerManager::HandlerType **HandlerType**

Public Member Functions

- [HandlerManager](#) (std::string, bool, bool)
Funcion que inicializa la clase.
- void [handle](#) (HttpRequest &hmsg)
Funcion que interpreta las httprequests.

Protected Attributes

- std::vector< [EventHandler](#) * > [handlers](#)
Variable vector con todos los handlers.
- [Database](#) * [db](#)
Variable puntero a Database.
- [TokenAuthenticator](#) * [auth](#)
Variable puntero a TokenAuthenticator.

3.15.1 Detailed Description

Clase que maneja los handlers.

Posee una lista con los handlers

3.15.2 Constructor & Destructor Documentation

3.15.2.1 HandlerManager::HandlerManager (std::string db_path, bool create_if_missing, bool clean_on_start)

Funcion que inicializa la clase.

Con los parametros que recibe crea un objeto [DatabaseLevelDB](#). Luego crea un [TokenAuthenticator](#) y con ambos llena la lista de handlers. EL NUMERO DE HANDLERS TIENE QUE COINCIDIR CON LOS ENMUS EN EL .H

3.15.3 Member Function Documentation

3.15.3.1 void HandlerManager::handle (HttpRequest & hmsg)

Funcion que interpreta las httprequests.

COLLECTION.

Dependiendo de los parametros que recibe llama al handler adecuado que lo tiene inicializado el vector handlers.

USERSELEMENT SESSIONS COLLECTION ELEMENT OTHER

The documentation for this class was generated from the following files:

- HandlerManager.h
- HandlerManager.cpp

3.16 HerokuComHandler Class Reference

Public Member Functions

- size_t **write_data** (char *ptr, size_t size, size_t nmemb, struct [url_data](#) *data)
- char * **GET** (char *url)
- char * **DELETE** (char *statham)
- char * **POST** (char *jason, char *statham)

Static Public Member Functions

- static size_t **matiMethodComeback** (char *data, size_t size, size_t nmemb, struct [url_data](#) *datax, void *p)

The documentation for this class was generated from the following files:

- HerokuComHandler.h
- HerokuComHandler.cpp

3.17 HttpRequest Class Reference

Clase que se encarga de manejar httprequests.

```
#include <HttpRequest.h>
```

Inheritance diagram for HttpRequest:

Public Types

- enum **UriType** { COLLECTION_URI, ELEMENT_URI }
- enum **MethodType** { POST, PUT, GET, DELETE, INVALID_METHOD }
- enum **UriField** { USERS, USERNAME, SESSIONS, SEARCH, PROFILE, METADATA, ADD_FRIEND, DELETE_FRIEND, ACCEPT_FRIEND, JOB, SKILL, LIKE, TOP, TOP_JOB, TOP_SKILL, INVALID_URI_FIELD }

- enum **StatusCode** {
OK = 200, **BAD_REQUEST** = 400, **UNAUTHORIZED** = 401, **NOT_FOUND** = 404,
CONFLICT = 409, **INTERNAL_ERROR** = 500 }
- typedef enum HttpRequest::UriType **UriType**
- typedef enum
 HttpRequest::MethodType **MethodType**
- typedef enum HttpRequest::UriField **UriField**
- typedef enum
 HttpRequest::StatusCode **StatusCode**

Public Member Functions

- [HttpRequest](#) ()
Funcion que inicializa la clase.
- virtual void [init](#) (struct mg_connection *n_conn, struct http_message *n_hmsg)=0
Establece los punteros nc y hmsg.
- virtual std::string [getUri](#) ()=0
Devuelve el uri.
- virtual void [getUriParsed](#) (std::vector< std::string > *parsed)=0
Funcion que parsea el uri en tokens.
- virtual UriField [getUriParsedByIndex](#) (int index)=0
Funcion que devuelve un uri de acuerdo a un indice.
- virtual int [getUriCantCampos](#) ()=0
Funcion que devuelve cuantos campos hay en el uri.
- virtual std::string [getUriStringParsedByIndex](#) (int index)=0
Funcion que devuelve un string con el uri de acuerdo a un indice.
- virtual UriType [getUriType](#) ()=0
Funcion que devuelve el tipo de uri.
- virtual MethodType [getMethod](#) ()=0
Funcion que devuelve el metodo de la request.
- virtual std::string [getCampo](#) (std::string)=0
Funcion que busca el campo pedido.
- virtual std::string [getCampoDeArray](#) (std::string campo, int index)=0
Funcion que busca el campo pedido del json.
- virtual std::string [getUsername](#) ()=0
Funcion que devuelve el username de la httprequest.
- virtual void [setResponse](#) (Status s, std::string r="")=0
Funcion que establece la respuesta.
- virtual std::string [getResponse](#) ()=0
Funcion que devuelve la variable response.
- virtual void [addValueToBody](#) (std::string name, std::string val)=0
Funcion que agrega valores a response.
- virtual unsigned int [getStatusCode](#) ()=0
Funcion que devuelve el codigo de status.
- virtual std::string [getConnToken](#) ()=0
Funcion que devuelve el conntoken.
- virtual std::string [getHeaderValue](#) (std::string name)=0
Funcion que devuelve el header que se corresponda con el string pasado.
- virtual std::string [getQueryCampo](#) (std::string)=0
Funcion que devuelve el campo query.

3.17.1 Detailed Description

Clase que se encarga de manejar httprequests.

Es una interface para las clases httprequestmock y httprequestconcrete.

The documentation for this class was generated from the following files:

- HttpRequest.h
- HttpRequest.cpp

3.18 HttpRequestConcrete Class Reference

Clase que maneja las httprequests.

```
#include <HttpRequestConcrete.h>
```

Inheritance diagram for HttpRequestConcrete:

Collaboration diagram for HttpRequestConcrete:

Public Member Functions

- [HttpRequestConcrete](#) ()
Funcion que inicializa la clase.
- virtual void [init](#) (struct mg_connection *n_conn, struct http_message *n_hmsg)
Establece los punteros nc y hmsg.
- virtual std::string [getUri](#) ()
Devuelve el uri.
- virtual void [getUriParsed](#) (std::vector< std::string > *parsed)
Funcion que parsea el uri en tokens.
- virtual int [getUriCantCampos](#) ()
Funcion que devuelve cuantos campos hay en el uri.
- virtual UriField [getUriParsedByIndex](#) (int index)
Funcion que devuelve un uri de acuerdo a un indice.
- virtual std::string [getUriStringParsedByIndex](#) (int index)
Funcion que devuelve un string con el uri de acuerdo a un indice.
- virtual UriType [getUriType](#) ()
Funcion que devuelve el tipo de uri.
- virtual MethodType [getMethod](#) ()
Funcion que devuelve el metodo de la request.
- virtual std::string [getCampo](#) (std::string)
Funcion que busca el campo pedido.
- virtual std::string [getCampoDeArray](#) (std::string campo, int index)
Funcion que busca el campo pedido del json.
- virtual std::string [getUsername](#) ()
Funcion que devuelve el username de la httprequest.
- virtual void [setResponse](#) (Status s, std::string r="")
Funcion que establece la respuesta.
- virtual std::string [getResponse](#) ()
Funcion que devuelve la variable response.
- virtual void [addValueToBody](#) (std::string name, std::string val)
Funcion que agrega valores a response.
- virtual unsigned int [getStatusCode](#) ()

Funcion que devuelve el codigo de status.

- virtual std::string [getConnToken](#) ()

Funcion que devuelve el conntoken.

- virtual std::string [getHeaderValue](#) (std::string name)

Funcion que devuelve el header que se corresponda con el string pasado.

- virtual std::string [getQueryCampo](#) (std::string)

Funcion que devuelve el campo query.

Additional Inherited Members

3.18.1 Detailed Description

Clase que maneja las httprequests.

3.18.2 Constructor & Destructor Documentation

3.18.2.1 [HttpRequestConcrete::HttpRequestConcrete](#) ()

Funcion que inicializa la clase.

Comienza con los punteros en NULL, ressource en "" y Status en ERROR.

3.18.3 Member Function Documentation

3.18.3.1 [void HttpRequestConcrete::addValueToBody](#) (std::string *name*, std::string *val*) [virtual]

Funcion que agrega valores a response.

Recibe dos string con nombre y valor y los agrega a la variable response.

Implements [HttpRequest](#).

Reimplemented in [HttpRequestMock](#).

3.18.3.2 [std::string HttpRequestConcrete::getQueryCampo](#) (std::string *name*) [virtual]

Funcion que devuelve el campo query.

Recibe un string y a partir de este devuelve un campo especifico de la httprequest.

Implements [HttpRequest](#).

Reimplemented in [HttpRequestMock](#).

3.18.3.3 [std::string HttpRequestConcrete::getUri](#) () [virtual]

Devuelve el uri.

Genera el uri y lo devuelve.

Implements [HttpRequest](#).

Reimplemented in [HttpRequestMock](#).

3.18.3.4 HttpRequestConcrete::UriType HttpRequestConcrete::getUriType () [virtual]

Funcion que devuelve el tipo de uri.

se fija si se trata de un COLLECTION_URI o un ELEMENT_URI.

Implements [HttpRequest](#).

Reimplemented in [HttpRequestMock](#).

3.18.3.5 void HttpRequestConcrete::setResponse (Status s, std::string r = " ") [virtual]

Funcion que establece la respuesta.

Recibe un status e inicializa la variable response en "", a la cual le agrega luego el estado.

Implements [HttpRequest](#).

The documentation for this class was generated from the following files:

- [HttpRequestConcrete.h](#)
- [HttpRequestConcrete.cpp](#)

3.19 HttpRequestMock Class Reference

Clase que maneja las httprequests.

```
#include <HttpRequestMock.h>
```

Inheritance diagram for HttpRequestMock:

Collaboration diagram for HttpRequestMock:

Public Member Functions

- void [init](#) (struct mg_connection *n_conn, struct http_message *n_hmsg)
Establece los punteros nc y hmsg.
- std::string [getUri](#) ()
Devuelve el uri.
- void [getUriParsed](#) (std::vector< std::string > &parsed)
Funcion que parsea el uri en tokens.
- UriField [getUriParsedByIndex](#) (int index)
Funcion que devuelve un uri de acuerdo a un indice.
- std::string [getUriStringParsedByIndex](#) (int index)
Funcion que devuelve un string con el uri de acuerdo a un indice.
- UriType [getUriType](#) ()
Funcion que devuelve el tipo de uri.
- MethodType [getMethod](#) ()
Funcion que devuelve el metodo de la request.
- std::string [getCampo](#) (std::string)
Funcion que busca el campo pedido.
- std::string [getCampoDeArray](#) (std::string campo, int index)
Funcion que busca el campo pedido del json.
- void [addValueToBody](#) (std::string name, std::string val)
Funcion que devuelve la respuesta.
- std::string [getHeaderValue](#) (std::string name)

Funcion que devuelve el codigo de status.

- `std::string` [getQueryCampo](#) (`std::string`)

Funcion que devuelve el campo query.

- [HttpRequestMock](#) ()

Funcion que inicializa la clase.

- [HttpRequestMock](#) (`std::string u`, `std::string t`)

Funcion que inicializa la clase.

- `std::string` [getUsername](#) ()

Funcion que devuelve el username de la httprequest.

- `std::string` [getConnToken](#) ()

Funcion que devuelve el conn token de la httprequest.

Public Attributes

- `std::string` [mock_username](#)

Variable string con username.

- `std::string` [mock_token](#)

Variable string con token.

- `std::string` **mock_ubicacion**
- `std::string` **mock_content**
- `std::vector< std::string >` **mock_add_tags**
- `std::vector< std::string >` **mock_add_write_perm**
- `std::vector< std::string >` **mock_remove_write_perm**
- `std::vector< std::string >` **mock_add_read_perm**
- `std::vector< std::string >` **mock_remove_read_perm**

Additional Inherited Members

3.19.1 Detailed Description

Clase que maneja las httprequests.

Sirve como mock para correr pruebas sin leveledb.

3.19.2 Constructor & Destructor Documentation

3.19.2.1 `HttpRequestMock::HttpRequestMock (std::string u, std::string t)` `[inline]`

Funcion que inicializa la clase.

Recibe dos string y los asigna a `mock_username` y a `mock_token`.

3.19.3 Member Function Documentation

3.19.3.1 `void HttpRequestMock::addValueToBody (std::string name, std::string val)` `[inline]`, `[virtual]`

Funcion que devuelve la respuesta.

Funcion que agrega valores a response.

Reimplemented from [HttpRequestConcrete](#).

3.19.3.2 `std::string HttpRequestMock::getHeaderValue (std::string name)` `[inline]`, `[virtual]`

Funcion que devuelve el codigo de status.

Funcion que devuelve el header que se corresponda con el string pasado.

Reimplemented from [HttpRequestConcrete](#).

The documentation for this class was generated from the following files:

- `HttpRequestMock.h`
- `HttpRequestMock.cpp`

3.20 JobGetHandler Class Reference

Clase handler que obtiene un usuario.

```
#include <JobGetHandler.h>
```

Inheritance diagram for JobGetHandler:

Collaboration diagram for JobGetHandler:

Public Member Functions

- [JobGetHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)

Funcion que inicializa la clase.

Additional Inherited Members

3.20.1 Detailed Description

Clase handler que obtiene un usuario.

Hereda de [EventHandlerChecksAuthentication](#).

3.20.2 Constructor & Destructor Documentation

3.20.2.1 `JobGetHandler::JobGetHandler (Database * db, TokenAuthenticator * a)`

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) al divino boton pero queda poco tiempo

The documentation for this class was generated from the following files:

- `JobGetHandler.h`
- `JobGetHandler.cpp`

3.21 JsonSerializer Class Reference

Clase que se encarga de tratar los archivos en formato json.

```
#include <JsonSerializer.h>
```

Public Member Functions

- void [turnObjectListToObject](#) (std::string &lista)
Funcion que pasa de lista de objetos a objeto.
- void [turnObjectListToObject](#) (std::string &lista, std::string name)
Funcion que pasa de lista de objetos a objeto bajo un nombre.
- void [turnArrayListToArray](#) (std::string &lista)
Funcion que pasa de lista de arrays a un array.
- void [turnArrayListToArray](#) (std::string &lista, std::string name)
Funcion que pasa de lista de arrays a un array bajo un nombre.
- void [addValueToObjectList](#) (std::string &lista, std::string name, std::string value)
Funcion que agrega un valor a una lista de objetos.
- void [addValueToObject](#) (std::string &obj, std::string name, std::string value)
Funcion que agrega un valor a un objeto.
- void [addValueToArrayList](#) (std::string &lista, std::string value)
Funcion que agrega un valor a una lista de arrays.
- void [joinValueIntoList](#) (std::string &lista, std::string value)
Funcion que une un valor con una lista.
- void [addArrayOrObjectToList](#) (std::string &lista, std::string name, std::string value)
Funcion que agrega un array u objeto a una lista.
- void [addNumberToList](#) (std::string &lista, std::string name, int number)
Funcion que agrega un numero a una lista.
- void [addBoolToList](#) (std::string &lista, std::string name, bool boolean)
Funcion que agrega un booleano a una lista.
- void [turnVectorToArray](#) (std::vector< std::string > &vec, std::string name, std::string &json)
Funcion que transforma un vector en un array.
- void [turnPairsVectorToObject](#) (std::vector< std::pair< std::string, std::string > > &vec, std::string name, std::string &json)
Funcion que transforma un vector de a pares en un objeto.
- void [turnMapToObj](#) (std::unordered_map< std::string, std::string > &mapa, std::string name, std::string &json)
Funcion que transforma un mapa en un objeto.

Static Public Member Functions

- static std::string [get](#) (Value value, std::string key, std::string default_val, Value &val, std::string &str_val)
Funcion get para un objeto.
- static std::string [get](#) (Value value, int key, std::string default_val, Value &val, std::string &str_val)
Funcion get para un array.
- static std::string [removeBegAndEndQuotes](#) (std::string)
Funcion que saca las comillas.

3.21.1 Detailed Description

Clase que se encarga de tratar los archivos en formato json.

Tiene diversas funciones para transformar los distintos formatos que pueden tomar las variables del json que vamos a utilizar.

3.21.2 Member Function Documentation

3.21.2.1 void JsonSerializer::addArrayOfObjectToList (std::string & *lista*, std::string *name*, std::string *value*)

Funcion que agrega un array u objeto a una lista.

Toma una lista y un value tipo: '["", "", ""]' o tipo '{ "", "", "" }' y los une en lista tipo: 'lista, "name" : value'.

3.21.2.2 void JsonSerializer::addBoolToList (std::string & *lista*, std::string *name*, bool *boolean*)

Funcion que agrega un booleano a una lista.

Lo mismo que addValueToObjectList pero sin agregarle comillas al valor 'true' o 'false' (si al nombre).

3.21.2.3 void JsonSerializer::addNumberToList (std::string & *lista*, std::string *name*, int *number*)

Funcion que agrega un numero a una lista.

Lo mismo que addValueToObjectList pero sin agregarle comillas al numero (si al nombre).

3.21.2.4 void JsonSerializer::addValueToArrayList (std::string & *lista*, std::string *value*)

Funcion que agrega un valor a una lista de arrays.

Toma una lista tipo: "pedro", "juan" y un value tipo: "jose" y lo transforma en lista en "pedro", "juan", "jose".

3.21.2.5 void JsonSerializer::addValueToObject (std::string & *obj*, std::string *name*, std::string *value*)

Funcion que agrega un valor a un objeto.

toma &obj de la forma { "" : "", "" : "" } y le agrega un campo "name" : "value".

3.21.2.6 void JsonSerializer::addValueToObjectList (std::string & *lista*, std::string *name*, std::string *value*)

Funcion que agrega un valor a una lista de objetos.

toma una lista tipo: "nombre1" : "pedro", "nombre2" : "juan" y un value tipo: "nombre3" : "jose" y lo transforma en lista en "nombre1" : "pedro", "nombre2" : "juan", "nombre3" : "jose".

3.21.2.7 std::string JsonSerializer::get (Value *value*, std::string *key*, std::string *default_val*, Value & *val*, std::string & *str_val*) [static]

Funcion get para un objeto.

Hace el get de JsonCpp en value (debe ser object), y guarda el resultado como value y string en val y str_val si era una hoja (es decir un valor tipo "nombre" : "pedro") en str_val se guarda: 'pedro' (sin comillas) si no encuentra la clave, devuelve default_val

3.21.2.8 std::string JsonSerializer::get (Value *value*, int *key*, std::string *default_val*, Value & *val*, std::string & *str_val*) [static]

Funcion get para un array.

Hace el get de JsonCpp en value (debe ser array), y guarda el resultado como value y string en val y str_val si era una hoja (es decir un valor tipo "nombre" : "pedro") en str_val se guarda: 'pedro' (sin comillas) si no encuentra el indice, devuelve default_val.

3.21.2.9 void JsonSerializer::joinValueIntoList (std::string & lista, std::string value)

Funcion que une un valor con una lista.

Toma una lista y un value y los une en lista tipo: 'lista, value'.

3.21.2.10 std::string JsonSerializer::removeBegAndEndQuotes (std::string str) [static]

Funcion que saca las comillas.

Saca las "" de los valores que se agarran de jsoncpp.

3.21.2.11 void JsonSerializer::turnArrayListToArray (std::string & lista)

Funcion que pasa de lista de arrays a un array.

toma una lista tipo: "pedro", "juan" y la convierte en lista en '["pedro", "juan"]'.

3.21.2.12 void JsonSerializer::turnArrayListToArray (std::string & lista, std::string name)

Funcion que pasa de lista de arrays a un array bajo un nombre.

toma una lista tipo: "pedro", "juan" y la convierte en lista en "@name" : ["pedro", "juan"]'.

3.21.2.13 void JsonSerializer::turnMapToObj (std::unordered_map< std::string, std::string > & mapa, std::string name, std::string & json)

Funcion que transforma un mapa en un objeto.

Recibe mapa, itera por sus elementos y arma un obj de la forma: { "key1" : "elem1", "key2" : "elem2", "key3" : "elem3" }.

3.21.2.14 void JsonSerializer::turnObjectListToObject (std::string & lista)

Funcion que pasa de lista de objetos a objeto.

Toma una lista tipo: "nombre1" : "pedro", "nombre2" : "juan" y la convierte en lista en '{ "nombre1" : "pedro", "nombre2" : "juan" }'.

3.21.2.15 void JsonSerializer::turnObjectListToObject (std::string & lista, std::string name)

Funcion que pasa de lista de objetos a objeto bajo un nombre.

Toma una lista tipo: "nombre1" : "pedro", "nombre2" : "juan" y la convierte en lista en "@name" : { "nombre1" : "pedro", "nombre2" : "juan" }'.

3.21.2.16 void JsonSerializer::turnPairsVectorToObject (std::vector< std::pair< std::string, std::string > > & vec, std::string name, std::string & json)

Funcion que transforma un vector de a pares en un objeto.

Recibe vec, itera por sus elementos y arma un array de la forma: { "nombre1" : "elem1", "nombre2" : "elem2", "nombre3" : "elem3" }.

3.21.2.17 void JsonSerializer::turnVectorToArray (std::vector< std::string > & vec, std::string name, std::string & json)

Funcion que transforma un vector en un array.

Recibe vec, itera por sus elementos y arma un array de la forma: ["elem1", "elem2", "elem3"].

The documentation for this class was generated from the following files:

- JsonSerializer.h
- JsonSerializer.cpp

3.22 LikeHandler Class Reference

Clase handler que elimina usuario.

```
#include <LikeHandler.h>
```

Inheritance diagram for LikeHandler:

Collaboration diagram for LikeHandler:

Public Member Functions

- [LikeHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)

Funcion que inicializa la clase.

Additional Inherited Members

3.22.1 Detailed Description

Clase handler que elimina usuario.

Hereda de [EventHandlerIgnoresAuthentication](#).

3.22.2 Constructor & Destructor Documentation

3.22.2.1 [LikeHandler::LikeHandler](#) ([Database](#) * db, [TokenAuthenticator](#) * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- LikeHandler.h
- LikeHandler.cpp

3.23 LoginHandler Class Reference

Clase handler que se encarga del login.

```
#include <LogInHandler.h>
```

Inheritance diagram for LoginHandler:

Collaboration diagram for LoginHandler:

Public Member Functions

- [LoginHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)

Funcion que inicializa la clase.

Additional Inherited Members

3.23.1 Detailed Description

Clase handler que se encarga del login.

Hereda de [EventHandlerGrantsAuthentication](#).

3.23.2 Constructor & Destructor Documentation

3.23.2.1 LoginHandler::LoginHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- LoginHandler.h
- LoginHandler.cpp

3.24 LogoutHandler Class Reference

Clase handler se encarga del logout.

```
#include <LogoutHandler.h>
```

Inheritance diagram for LogoutHandler:

Collaboration diagram for LogoutHandler:

Public Member Functions

- [LogoutHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)

Funcion que inicializa la clase.

Additional Inherited Members

3.24.1 Detailed Description

Clase handler se encarga del logout.

Hereda de [EventHandlerRemovesAuthentication](#).

3.24.2 Constructor & Destructor Documentation

3.24.2.1 LogoutHandler::LogoutHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- [LogOutHandler.h](#)
- [LogOutHandler.cpp](#)

3.25 RespondFriendRequestHandler Class Reference

Clase handler que responde notificacion de agregar amigo.

```
#include <RespondFriendRequestHandler.h>
```

Inheritance diagram for RespondFriendRequestHandler:

Collaboration diagram for RespondFriendRequestHandler:

Public Member Functions

- [RespondFriendRequestHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)
Funcion que inicializa la clase.

Additional Inherited Members

3.25.1 Detailed Description

Clase handler que responde notificacion de agregar amigo.

Hereda de [EventHandlerChecksAuthentication](#).

3.25.2 Constructor & Destructor Documentation

3.25.2.1 RespondFriendRequestHandler::RespondFriendRequestHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- [RespondFriendRequestHandler.h](#)
- [RespondFriendRequestHandler.cpp](#)

3.26 SendFriendRequestHandler Class Reference

Clase handler que envia notificacion de agregar amigo.

```
#include <SendFriendRequestHandler.h>
```

Inheritance diagram for SendFriendRequestHandler:

Collaboration diagram for SendFriendRequestHandler:

Public Member Functions

- [SendFriendRequestHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)
Funcion que inicializa la clase.

Additional Inherited Members

3.26.1 Detailed Description

Clase handler que envia notificacion de agregar amigo.

Hereda de [EventHandlerChecksAuthentication](#).

3.26.2 Constructor & Destructor Documentation

3.26.2.1 `SendFriendRequestHandler::SendFriendRequestHandler (Database * db, TokenAuthenticator * a)`

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- `SendFriendRequestHandler.h`
- `SendFriendRequestHandler.cpp`

3.27 Server Class Reference

Clase que maneja el server.

```
#include <Server.h>
```

Public Member Functions

- [Server](#) (std::string, bool, bool)
Funcion que inicializa la clase.
- void [poll](#) ()
Funcion que comienza el chequeo de llegadas de mensajes.
- void [stop](#) ()
Funcion que para el chequeo de llegadas de mensajes.
- bool [createListeningConnection](#) (const char *addr)
Funcion que establece la conexion para recibir mensajes.

3.27.1 Detailed Description

Clase que maneja el server.

3.27.2 Constructor & Destructor Documentation

3.27.2.1 `Server::Server (std::string db_path, bool create_if_missing, bool clean_on_start)`

Funcion que inicializa la clase.

Con el string y los bool que recibe crea un [HandlerManager](#). Luego crea una estructura mg_mgr y establece active en FALSE.

3.27.3 Member Function Documentation

3.27.3.1 bool Server::createListeningConnection (const char * *addr*)

Funcion que establece la conexion para recibir mensajes.

Recibe la direccion a donde conectarse y con mongoose establece la conexion.

3.27.3.2 void Server::poll ()

Funcion que comienza el chequeo de llegadas de mensajes.

Llama a la funcion poll de mongoose.

3.27.3.3 void Server::stop ()

Funcion que para el chequeo de llegadas de mensajes.

Llama a la funcion free de mongoose.

The documentation for this class was generated from the following files:

- Server.h
- Server.cpp

3.28 Server_Logger Class Reference

Clase que maneja el logger.

```
#include <Logger.h>
```

Public Member Functions

- void [setLogPath](#) (std::string path)
Funcion que establece el logpath.
- void [Log](#) (std::string comment, LOG_MODE mode)
Funcion que genera el log.

Static Public Member Functions

- static [Server_Logger](#) * [getInstance](#) ()
Funcion que devuelve la instancia del logger.

3.28.1 Detailed Description

Clase que maneja el logger.

3.28.2 Member Function Documentation

3.28.2.1 [Server_Logger](#) * [Server_Logger::getInstance](#) () [static]

Funcion que devuelve la instancia del logger.

Si no la encuentra la crea.

The documentation for this class was generated from the following files:

- [Logger.h](#)
- [Logger.cpp](#)

3.29 SignUpHandler Class Reference

Clase handler que se encarga del signup.

```
#include <SignUpHandler.h>
```

Inheritance diagram for SignUpHandler:

Collaboration diagram for SignUpHandler:

Public Member Functions

- [SignUpHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)
Funcion que inicializa la clase.

Additional Inherited Members

3.29.1 Detailed Description

Clase handler que se encarga del signup.

Hereda de [EventHandlerIgnoresAuthentication](#).

3.29.2 Constructor & Destructor Documentation

3.29.2.1 SignUpHandler::SignUpHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- [SignUpHandler.h](#)
- [SignUpHandler.cpp](#)

3.30 SkillGetHandler Class Reference

Clase handler que obtiene un usuario.

```
#include <SkillGetHandler.h>
```

Inheritance diagram for SkillGetHandler:

Collaboration diagram for SkillGetHandler:

Public Member Functions

- [SkillGetHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)
Funcion que inicializa la clase.

Additional Inherited Members

3.30.1 Detailed Description

Clase handler que obtiene un usuario.

Hereda de [EventHandlerChecksAuthentication](#).

3.30.2 Constructor & Destructor Documentation

3.30.2.1 SkillGetHandler::SkillGetHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) al divino boton pero queda poco tiempo

The documentation for this class was generated from the following files:

- SkillGetHandler.h
- SkillGetHandler.cpp

3.31 TokenAuthenticator Class Reference

Clase que se encarga de manejar tokens.

```
#include <TokenAuthenticator.h>
```

Public Member Functions

- [TokenAuthenticator](#) ()
Funcion que inicializa la clase.
- bool [isValidToken](#) (std::string user, std::string token)
Funcion que valida tokens.
- std::string [createToken](#) (std::string user)
Funcion que agrega tokens.
- bool [removeToken](#) (std::string user)
Funcion que remueve tokens.

3.31.1 Detailed Description

Clase que se encarga de manejar tokens.

Esta clase crea, valida y destruye tokens empleando una clase map.

3.31.2 Member Function Documentation

3.31.2.1 std::string TokenAuthenticator::createToken (std::string user)

Funcion que agrega tokens.

Recibe un user y devuelve un token.

3.31.2.2 bool TokenAuthenticator::isValidToken (std::string user, std::string token)

Funcion que valida tokens.

Esta función recibe un user y un token y busca en su map si se encuentran. log->Log("username : ", user, TRACE);
log->Log("token : %s", token, TRACE);

3.31.2.3 bool TokenAuthenticator::removeToken (std::string user)

Funcion que remueve tokens.

Recibe el user a borrar y lo elimina de su map.

The documentation for this class was generated from the following files:

- TokenAuthenticator.h
- TokenAuthenticator.cpp

3.32 TopJobHandler Class Reference

Clase handler que obtiene un usuario.

```
#include <TopJobHandler.h>
```

Inheritance diagram for TopJobHandler:

Collaboration diagram for TopJobHandler:

Public Member Functions

- [TopJobHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)

Funcion que inicializa la clase.

Additional Inherited Members

3.32.1 Detailed Description

Clase handler que obtiene un usuario.

Hereda de [EventHandlerChecksAuthentication](#).

3.32.2 Constructor & Destructor Documentation

3.32.2.1 TopJobHandler::TopJobHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) al divino boton pero queda poco tiempo

The documentation for this class was generated from the following files:

- TopJobHandler.h
- TopJobHandler.cpp

3.33 TopRankGetHandler Class Reference

Clase handler que obtiene un usuario.

```
#include <TopRankGetHandler.h>
```

Inheritance diagram for TopRankGetHandler:

Collaboration diagram for TopRankGetHandler:

Public Member Functions

- [TopRankGetHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)

Funcion que inicializa la clase.

Additional Inherited Members

3.33.1 Detailed Description

Clase handler que obtiene un usuario.

Hereda de [EventHandlerChecksAuthentication](#).

3.33.2 Constructor & Destructor Documentation

3.33.2.1 TopRankGetHandler::TopRankGetHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) al divino boton pero queda poco tiempo

The documentation for this class was generated from the following files:

- TopRankGetHandler.h
- TopRankGetHandler.cpp

3.34 url_data Struct Reference

Public Attributes

- `size_t` **size**
- `char *` **data**

The documentation for this struct was generated from the following file:

- HerokuComHandler.h

3.35 User Class Reference

Clase que maneja al usuario.

```
#include <User.h>
```

Inheritance diagram for User:

Collaboration diagram for User:

Public Member Functions

- [User](#) ([Database](#) *db, [DatabaseWriteBatch](#) *dbbatch=NULL)
Funcion que inicializa la clase.
- void [setUsername](#) (std::string usr)
Funcion que establece el username.
- void [setPassword](#) (std::string pass)
Funcion que establece la password.
- std::string [getUsername](#) ()
Funcion que retorna el username.
- std::string [getPassword](#) ()
Funcion que retorna la password.
- Status [DBerase](#) ()
Funcion que elimina al user de la base de datos.
- Status [DBget](#) ()
Funcion que devuelve el estado de su busqueda en la base de datos.
- Status [DBcreate](#) ()
Funcion que crea al usuario en la base de datos.

Protected Member Functions

- virtual void [_setKey](#) ()
Funcion que asigna username a la key.
- virtual void [_setValue](#) ()
Funcion que asigna password al value.
- virtual void [_setValueVars](#) ()
Funcion que asigna value a la password.

Additional Inherited Members

3.35.1 Detailed Description

Clase que maneja al usuario.

Esta clase hereda de [DElement](#).

3.35.2 Constructor & Destructor Documentation

3.35.2.1 [User::User](#) ([Database](#) * db, [DatabaseWriteBatch](#) * dbbatch = NULL)

Funcion que inicializa la clase.

Esta funcion emplea la misma inicializacion que [DBElement](#).

3.35.3 Member Function Documentation

3.35.3.1 virtual void [User::_setKey](#) () [inline], [protected], [virtual]

Funcion que asigna username a la key.

De esta forma se puede manejar el elemento a traves de funciones definidas en la clase madre.

Implements [DBElement](#).

3.35.3.2 `virtual void User::_setValue () [inline], [protected], [virtual]`

Funcion que asigna password al value.

De esta forma se puede manejar el elemento a traves de funciones definidas en la clase madre.

Implements [DBElement](#).

3.35.3.3 `virtual void User::_setValueVars () [inline], [protected], [virtual]`

Funcion que asigna value a la password.

De esta forma se puede manejar el elemento a traves de funciones definidas en la clase madre.

Implements [DBElement](#).

3.35.3.4 `Status User::DBcreate ()`

Funcion que crea al usuario en la base de datos.

En caso de no existir previamente, crea un usuario con el username del objeto.

3.35.3.5 `Status User::DBget ()`

Funcion que devuelve el estado de su busqueda en la base de datos.

Se busca a si mismo en la base de datos y devuelve el status resultante de eso.

3.35.3.6 `void User::setPassword (std::string pass) [inline]`

Funcion que establece la password.

Recibe un string y lo asigna a password.

3.35.3.7 `void User::setUsername (std::string usr) [inline]`

Funcion que establece el username.

Recibe un string y lo asigna a username.

The documentation for this class was generated from the following files:

- User.h
- User.cpp

3.36 UserDeleteHandler Class Reference

Clase handler que elimina usuario.

```
#include <UserDeleteHandler.h>
```

Inheritance diagram for UserDeleteHandler:

Collaboration diagram for UserDeleteHandler:

Public Member Functions

- [UserDeleteHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)

Funcion que inicializa la clase.

Additional Inherited Members

3.36.1 Detailed Description

Clase handler que elimina usuario.

Hereda de [EventHandlerRemovesAuthentication](#).

3.36.2 Constructor & Destructor Documentation

3.36.2.1 UserDeleteHandler::UserDeleteHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- UserDeleteHandler.h
- UserDeleteHandler.cpp

3.37 UserGetHandler Class Reference

Clase handler que obtiene un usuario.

```
#include <UserGetHandler.h>
```

Inheritance diagram for UserGetHandler:

Collaboration diagram for UserGetHandler:

Public Member Functions

- [UserGetHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)

Funcion que inicializa la clase.

Additional Inherited Members

3.37.1 Detailed Description

Clase handler que obtiene un usuario.

Hereda de [EventHandlerChecksAuthentication](#).

3.37.2 Constructor & Destructor Documentation

3.37.2.1 UserGetHandler::UserGetHandler (Database * db, TokenAuthenticator * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- UserGetHandler.h
- UserGetHandler.cpp

3.38 UserGetProfileHandler Class Reference

Clase que devuelve el perfil de otros usuarios.

```
#include <UserGetProfileHandler.h>
```

Inheritance diagram for UserGetProfileHandler:

Collaboration diagram for UserGetProfileHandler:

Public Member Functions

- [UserGetProfileHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)
Funcion que inicializa la clase.

Additional Inherited Members

3.38.1 Detailed Description

Clase que devuelve el perfil de otros usuarios.

Ignora el token ya que deberian ser publicos los perfiles

3.38.2 Constructor & Destructor Documentation

3.38.2.1 UserGetProfileHandler::UserGetProfileHandler ([Database](#) * db, [TokenAuthenticator](#) * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- UserGetProfileHandler.h
- UserGetProfileHandler.cpp

3.39 UserMetadata Class Reference

Clase que contiene la informacion del usuario.

```
#include <UserMetadata.h>
```

Inheritance diagram for UserMetadata:

Collaboration diagram for UserMetadata:

Public Member Functions

- [UserMetadata](#) ([Database](#) *db, [DatabaseWriteBatch](#) *dbbatch=NULL)
Funcion que inicializa la clase.
- void [setUsername](#) (std::string name)
Funcion que establece el username.

- void [setJoinDate](#) (std::string date)
Funcion que establece la fecha de inscripcion.
- void [changeResume](#) (std::string n_email)
Funcion que cambia el resume por uno nuevo.
- void [changeEmail](#) (std::string n_email)
Funcion que cambia el mail por uno nuevo.
- std::string [getEmail](#) ()
- void [changeUltimaUbicacion](#) (std::string u)
Funcion que cambia la ultima ubicacion del usuario.
- void [changeJobList](#) (std::string u)
Funcion que cambia la lista de job.
- void [changeSkillList](#) (std::string u)
Funcion que cambia la lista de skill.
- void [addNotificacion](#) (std::string u)
Funcion que cambia la lista de notificaciones.
- void [removeNotification](#) (std::string u)
Funcion que borra una notificacion.
- void [addFriend](#) (std::string u)
Funcion que cambia la lista de contactos.
- void [addLike](#) (std::string u)
Funcion que cambia la lista de Likes.
- void [removeLike](#) (std::string u)
Funcion que borra un Like.
- void [removeFriend](#) (std::string u)
Funcion que borra un contacto.
- void [changeName](#) (std::string n)
Funcion que cambia el nombre del usuario.
- void [changePic](#) (std::string p)
Funcion que cambia la foto del usuario del usuario.
- std::string [getUltimaUbicacion](#) ()
- std::string [getJoinDate](#) ()
Funcion que devuelve la fecha de inscripcion.
- std::string [getUsername](#) ()
Funcion que devuelve el username.
- std::vector< std::string > * [getLikeList](#) ()
Funcion que devuelve el username.
- Status [DBget](#) ()
Funcion que obtiene la metadata del usuario de la base de datos.
- Status [DBchange_profile_pic](#) (std::string picture)
Funcion que cambia la foto de perfil.
- Status [DBchange_name](#) (std::string name)
Funcion que cambia el nombre que aparece en perfil.
- Status [DBchange_last_place](#) (std::string place)
Funcion que cambia la ultima ubicacion.
- Status [DBchange_email](#) (std::string n_email)
Funcion que cambia mail en la base de datos.
- Status [DBchange_resume](#) (std::string n_email)
Funcion que cambia resume en la base de datos.
- Status [DBchange_skill_list](#) (std::string n_email)
Funcion que cambia skill list en la base de datos.
- Status [DBchange_job_list](#) (std::string n_email)

- Funcion que cambia job list en la base de datos.*

 - Status [DBaddNotification](#) (std::string u)

Funcion que agrega una notificacion en la base de datos.

 - Status [DBremoveNotification](#) (std::string u)

Funcion que borra una notificacion en la base de datos.

 - Status [DBaddFriend](#) (std::string u, bool cyclic=false)

Funcion que agrega un amigo en la base de datos.

 - Status [DBremoveFriend](#) (std::string u)

Funcion que borra un amigo en la base de datos.

 - Status [DBaddLike](#) (std::string u)

Funcion que agrega un Like en la base de datos.

 - Status [DBremoveLike](#) (std::string u)

Funcion que borra un Like en la base de datos.

 - std::string [getProfile](#) ()

Funcion que devuelve el perfil del usuario.

Public Attributes

- std::string [jobList](#)
- Variable string que contiene jobList.*
- std::string [skillList](#)
- Variable string que contiene skillList.*
- std::vector< std::string > [likeList](#)
- Variable para la lista de Likes.*

Protected Member Functions

- virtual void [_setKey](#) ()
- Funcion que establece la key.*
- virtual void [_setValue](#) ()
- Funcion que establece el value.*
- virtual void [_setValueVars](#) ()
- Funcion que establece varias variables.*

Friends

- class **User**

Additional Inherited Members

3.39.1 Detailed Description

Clase que contiene la informacion del usuario.

Tiene los datos del usuario y se encarga de manejarlos\ Hereda de [DBElement](#).

3.39.2 Constructor & Destructor Documentation

3.39.2.1 UserMetadata::UserMetadata (Database * db, DatabaseWriteBatch * dbbatch = NULL)

Funcion que inicializa la clase.

Realiza la misma inicializacion que [DBElement](#), pero ademas sus variables particulares las inicia en "", 0 o en el caso de couta_max, en 10GB.

3.39.3 Member Function Documentation

3.39.3.1 void UserMetadata::_setKey () [protected],[virtual]

Funcion que establece la key.

Toma la variable username y la establece como key, luego de agrgarle ".usertoken".

Implements [DBElement](#).

3.39.3.2 void UserMetadata::_setValue () [protected],[virtual]

Funcion que establece el value.

Toma las variables join_date, email y forma un json.

Implements [DBElement](#).

3.39.3.3 void UserMetadata::_setValueVars () [protected],[virtual]

Funcion que establece varias variables.

Toma la variable value y a partir de ella obtiene join_date, email y ultima_ubicacion.

Implements [DBElement](#).

3.39.3.4 void UserMetadata::addFriend (std::string u) [inline]

Funcion que cambia la lista de contactos.

Recibe como string la lista de contactos y la asigna a la a la ultima .

3.39.3.5 void UserMetadata::addLike (std::string u) [inline]

Funcion que cambia la lista de Likes.

Recibe como string la lista de Likes y la asigna a la a la ultima .

3.39.3.6 void UserMetadata::addNotificacion (std::string u) [inline]

Funcion que cambia la lista de notificaciones.

Recibe como string las nuevas notificaciones y la asigna a la variable.

3.39.3.7 void UserMetadata::changeEmail (std::string n_email) [inline]

Funcion que cambia el mail por uno nuevo.

Recibe como string el nuevo mail y lo asigna a la variable.

3.39.3.8 void UserMetadata::changeJobList (std::string *u*) [inline]

Funcion que cambia la lista de job.

Recibe como string el nuevo job y la asigna a la variable.

3.39.3.9 void UserMetadata::changeName (std::string *n*) [inline]

Funcion que cambia el nombre del usuario.

Recibe el nombre y cambia su variable asociada

3.39.3.10 void UserMetadata::changePic (std::string *p*) [inline]

Funcion que cambia la foto del usuario del usuario.

Recibe la foto encriptada y cambia su variable asociada

3.39.3.11 void UserMetadata::changeResume (std::string *n_email*) [inline]

Funcion que cambia el resume por uno nuevo.

Recibe como string el nuevo resume y lo asigna a la variable.

3.39.3.12 void UserMetadata::changeSkillList (std::string *u*) [inline]

Funcion que cambia la lista de skill.

Recibe como string el nuevo skill y la asigna a la variable.

3.39.3.13 void UserMetadata::changeUltimaUbicacion (std::string *u*) [inline]

Funcion que cambia la ultima ubicacion del usuario.

Recibe como string la nueva ubicacion y la asigna a la variable.

3.39.3.14 std::string UserMetadata::getProfile ()

Funcion que devuelve el perfil del usuario.

Devuelve el perfil del usuario en formato json

3.39.3.15 void UserMetadata::removeFriend (std::string *u*)

Funcion que borra un contacto.

Recibe como string el nombre del contacto

3.39.3.16 void UserMetadata::removeLike (std::string *u*)

Funcion que borra un Like.

Recibe como string el nombre del Like

3.39.3.17 void UserMetadata::removeNotification (std::string *u*)

Funcion que borra una notificacion.

Recibe como string cual borrar.

3.39.3.18 void UserMetadata::setJoinDate (std::string *date*) [inline]

Funcion que establece la fecha de inscripcion.

Recibe un string y lo establece como la fecha.

3.39.3.19 void UserMetadata::setUsername (std::string *name*) [inline]

Funcion que establece el username.

Recibe un string y se lo asigna a username.

The documentation for this class was generated from the following files:

- UserMetadata.h
- UserMetadata.cpp

3.40 UserModifyHandler Class Reference

Clase handler que modifica un usuario.

```
#include <UserModifyHandler.h>
```

Inheritance diagram for UserModifyHandler:

Collaboration diagram for UserModifyHandler:

Public Member Functions

- [UserModifyHandler](#) ([Database](#) *db, [TokenAuthenticator](#) *a)
Funcion que inicializa la clase.

Additional Inherited Members

3.40.1 Detailed Description

Clase handler que modifica un usuario.

Hereda de [EventHandlerChecksAuthentication](#).

3.40.2 Constructor & Destructor Documentation

3.40.2.1 UserModifyHandler::UserModifyHandler ([Database](#) * db, [TokenAuthenticator](#) * a)

Funcion que inicializa la clase.

Recibe punteros a [Database](#) y a [TokenAuthenticator](#) y los guarda en db y auth.

The documentation for this class was generated from the following files:

- UserModifyHandler.h

- UserModifyHandler.cpp

Index

- close
 - Database, [6](#)
- config
 - Database, [6](#)
- Database, [5](#)
 - close, [6](#)
 - config, [6](#)
 - Database, [6](#)
 - erase, [6](#)
 - get, [6](#)
 - open, [6](#)
 - put, [6](#)
- erase
 - Database, [6](#)
- get
 - Database, [6](#)
- open
 - Database, [6](#)
- poll
 - Server, [39](#)
- put
 - Database, [6](#)
- Server, [38](#)
 - poll, [39](#)
 - Server, [38](#)
 - stop, [39](#)
- stop
 - Server, [39](#)
- User, [43](#)
 - User, [44](#)