Class Documentation

AnswerCallback

Detailed Description

Answer callback - an answer callback.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

AppSocket

Protected Attributes

- final Socket **msgSocket**

Detailed Description

App socket - represents a communications socket able to send and receive messages from the client to the server and vice versa.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

AppSocket (String *ip*, int *port*) throws IOException

Instantiates a new App socket.

Parameters

ip	the ip
port	the port

Exceptions

IOException	the io exception

AppSocket (Socket *socket*) throws IOException

Instantiates a new App socket.

Parameters

socket	the socket

Exceptions

IOException	the io exception

Member Function Documentation

void close ()

Close.

void close (MyError *err*)

Close.

Parameters

err	the error

String getLocalAddress ()

Gets local address.

Returns

the local address

MessagesHandler getMessagesHandler ()

Gets messages handler.

Returns

the messages handler

String getRemoteAddress ()

Gets remote address.

Returns

the remote address

void handledRun () `[protected]`

Handled run.

Reimplemented from **MyThread** (*p.95*).

void interruptListener (MyError *err*)

Interrupt listener.

Parameters

err	the err to interrupt with

boolean isClosed ()

Is closed boolean.

Returns

the boolean

boolean isConnected ()

Is connected boolean.

Returns

the boolean

Message requestMessage (Message *requestMsg*)

sending request and blocking til res

Parameters

requestMsg	= "can i have x message?"

Returns

response

void requestMessage (Message *requestMsg*, MessageCallback *onRes*)

Request message.

Parameters

requestMsg	the request msg
onRes	the on res

void respond (Message *msg*, Message *respondingTo*)

Respond.

Parameters

msg	the msg
respondingTo	the responding to

void setMessagesHandler (MessagesHandler *messagesHandler*)

Sets messages handler.

Parameters

messagesHandler	the messages handler

void stopReading ()

Stop reading.

synchronized void writeMessage (Message *msg*)

Write message.

Parameters

msg	the msg

Member Data Documentation

final Socket msgSocket `[protected]`

The Msg socket.

ArchivedGameInfo

Additional Inherited Members

Detailed Description

Archived game info.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

ArchivedGameInfo (String *gameId*, String *creatorUsername*, String *opponentUsername*, GameSettings *gameSettings*, String *winner*, Stack< Move > *moveStack*)

Instantiates a new Archived game info.

Parameters

gameId	the game id
creatorUsername	the creator username
opponentUsername	the opponent username
gameSettings	the game settings
winner	the winner
moveStack	the move stack

Member Function Documentation

String getWinner ()

Gets winner.

Returns

the winner

String toString ()

To string string.

Returns

the string

Reimplemented from **EstablishedGameInfo** (*p.41*).

Arg

Public Attributes

- final String **repInStr**
- final boolean **escape**
- final **ArgType argType**
- final **Config**<?> **config**

Detailed Description

Arg - .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Arg (ArgType *argType*)

Instantiates a new **Arg**.

Parameters

argType	the arg type

Arg (ArgType *argType*, Config<?> *config*)

Instantiates a new **Arg**.

Parameters

argType	the arg type
config	the config

Arg (ArgType *argType*, boolean *escape*, Config<?> *config*)

Instantiates a new **Arg**.

Parameters

argType	the arg type
escape	the escape
config	the config

Member Function Documentation

String createVal (Object *val*)

Create val string.

Parameters

val	the val

Returns

the string

boolean equals (Object *o*)

Equals boolean.

Parameters

o	the o

Returns

the boolean

boolean isUserInput ()

Is user input boolean.

Returns

the boolean

void setUserInput (boolean *userInput*)

Sets user input.

Parameters

userInput	the user input

String toString ()

To string string.

Returns

the string

Member Data Documentation

final ArgType argType

The **Arg** type.

final Config<?> config

The **Config**.

final boolean escape

The Escape.

final String repInStr

The Rep in str.

ArgsUtil

Detailed Description

The type Args utils.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

static ArgsUtil create (String[] *args*)`[static]`

Create args util.

Parameters

args	the args

Returns

the args util

OptionalArg equalsSign (String *preEqualStr*)

Equals sign optional arg. for any arg of this format: preEqualStr=argval%

Parameters

preEqualStr	the pre equal str

Returns

the optional arg value(assuming there is one) argval% in the example above

record OptionalArg (String *str*)

Optional arg.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

OptionalArg plainTextIgnoreCase (String *str*)

Plain text ignore case optional arg.

Parameters

str	the str

Returns

the optional arg

ArgType Enum Reference

Public Attributes

- **Date**
- **DateRange**
- **Text**
- **ServerAddress**
- **Number**
- **Username**
- **Password**
- **Url**
- **PictureUrl**
- final boolean **isUserInput**

Detailed Description

Arg type - argument type.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

ArgType ()

Instantiates a new **Arg** type.

ArgType (boolean *isUserInput*)

Instantiates a new **Arg** type.

Parameters

isUserInput	the is user input

Member Data Documentation

Date

Date arg type.

DateRange

Date range arg type.

final boolean isUserInput

The Is user input.

Number

Number arg type.

Password

Password arg type.

PictureUrl

Picture url arg type.

ServerAddress

Server address arg type.

Text

Text arg type.

Url

Url arg type.

Username

Username arg type.

ArrUtils

Detailed Description

Array Utility Class.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

static< T > T[] concat (T[] *array1*, T... *array2*) `[static]`

`credit`

Parameters

<T>	the type parameter
array1	the array 1
array2	the array 2

Returns

t [ ]

static< T > ArrayList< T > createList (ObjCallback< T > *objCreator*, int *size*) `[static]`

Create list array list.

Parameters

<T>	the type parameter
objCreator	the obj creator
size	the size

Returns

the array list

static< T > T exists (T[] *arr*, int... *index*) `[static]`

Exists t.

Parameters

<T>	the type parameter
arr	the arr
index	the index

Returns

the t

BasicMove

Detailed Description

Basic move - represents a basic move. with a source and a destination.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

BasicMove (BasicMove *other*)

Copy constructor.

Parameters

other	the other

BasicMove (Location *movingFrom*, Location *movingTo*)

Instantiates a new Basic move.

Parameters

movingFrom	the moving from
movingTo	the moving to

BasicMove (String *move*)

Instantiates a new Basic move.

Parameters

move	the move

Member Function Documentation

BasicMove cp ()

copies this move.

Returns

the new copy

static BasicMove[] createBatch (Location... *locs*)`[static]`

Create batch basic move [ ].

Parameters

locs	the locs

Returns

the basic move [ ]

boolean equals (Object *o*)

Equals boolean.

Parameters

o	the o

Returns

the boolean

Reimplemented in **Move** (*p.84*).

void flip ()

Flips the source and destination.

String getBasicMoveAnnotation ()

Gets basic move annotation.

Returns

the basic move annotation

static BasicMove getFlipped (BasicMove *basicMove*)`[static]`

Gets a copy of the provided move with the source and destination flipped

Parameters

basicMove	the original move

Returns

the flipped move

Location getMovingFrom ()

Gets moving from. aka source

Returns

the moving from

Location getMovingTo ()

Gets moving to.

Returns

the moving to

int hashCode ()

Hash code int.

Returns

the int

void setMovingFrom (Location *movingFrom*)

Sets moving from.

Parameters

movingFrom	the moving from

void setMovingTo (Location *movingTo*)

Sets moving to.

Parameters

movingTo	the moving to

String toString ()

To string string.

Returns

the string

Reimplemented in **Move** (*p.87*).

BitData

Static Public Attributes

- static final long **notAFile**
- static final long **notHFile**
- static final long **everything**

Detailed Description

Bit data - utility class meant for storing useful board constants. (in bitboard format of course)

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Data Documentation

final long everything `[static]`

The constant everything. the whole board

final long notAFile `[static]`

The constant notAFile. the whole board but the A file

final long notHFile `[static]`

The constant notHFile. the whole board but the H file

Board

Static Public Attributes

- static final String **startingFen**
- static final **Board example**

Detailed Description

Board - represents the logic board of the model

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Board (Board *other*)

Instantiates a new **Board**.

Parameters

other	the other

Board ()

Instantiates a new **Board**.

Board (String *fen*)

Instantiates a new **Board**.

Parameters

fen	the fen

Member Function Documentation

void fenSetup (String *fen*)

Fen setup.

Parameters

fen	the fen

Piece getPiece (Location *loc*)

Gets piece.

Parameters

loc	the loc

Returns

the piece

Piece getPiece (Location *loc*, boolean *notNull*)

Gets piece.

Parameters

loc	the loc
notNull	the not null

Returns

the piece

Square[] getRow (int *row*, boolean *flipLocs*)

Get row square [ ].

Parameters

row	the row
flipLocs	the flip locs

Returns

the square [ ]

Square getSquare (Location *loc*)

Gets square.

Parameters

loc	the loc

Returns

the square

boolean isSquareEmpty (Location *loc*)

Is square empty boolean.

Parameters

loc	the loc

Returns

the boolean

Iterator< Square > iterator ()

Iterator iterator.

Returns

the iterator

void print ()

Print.

void setPiece (Location *loc*, Piece *piece*)

Sets piece.

Parameters

loc	the loc
piece	the piece

void setSquareEmpty (Location *loc*)

Sets square empty.

Parameters

loc	the loc

static Board startingPos () `[static]`

Starting pos board.

Returns

the board

String toString ()

To string string.

Returns

the string

Member Data Documentation

final Board example `[static]`

The constant example.

final String startingFen `[static]`

The constant startingFen.

MyJFrame.BooleanClosing

Additional Inherited Members

Detailed Description

The interface Boolean closing.

Member Function Documentation

default boolean checkClosingVal (Boolean *val*)

Check closing val boolean.

Parameters

val	the val

Returns

the boolean

void closing ()

Closing.

default Boolean show ()

Show boolean.

Returns

the boolean

Callback< T >

Detailed Description

Callback - an object callback.

Parameters

<T>	the object's type

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

void callback (T *obj*)

Callback.

Parameters

obj	the obj

CastlingRights

Classes

- enum **Side**

Static Public Attributes

- static final String **NO_CASTLING_ABILITY**

Detailed Description

Castling rights - .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

CastlingRights (byte *rights*)

Instantiates a new Castling rights.

Parameters

rights	the rights

CastlingRights ()

Instantiates a new Castling rights.

CastlingRights (CastlingRights *other*)

Instantiates a new Castling rights.

Parameters

other	the other

Member Function Documentation

static CastlingRights createFromStr (String *castlingAbilityStr*) `[static]`

Create from str castling rights.

Parameters

castlingAbilityStr	the castling ability str

Returns

the castling rights

byte disableCastling (PlayerColor *playerColor*, Side *side*)

Disable castling byte.

Parameters

playerColor	the player color
side	the side

Returns

the byte

void enable (byte *b*)

Enable.

Parameters

b	the b

void enableCastling (PlayerColor *playerColor*, Side *side*)

Enable castling.

Parameters

playerColor	the player color
side	the side

byte getPlayersCastling (PlayerColor *playerColor*)

Gets players castling.

Parameters

playerColor	the player color

Returns

the players castling

byte getRights ()

Gets rights.

Returns

the rights

boolean hasAny (PlayerColor *playerColor*)

Has any boolean.

Parameters

playerColor	the player color

Returns

the boolean

boolean isEnabled (PlayerColor *playerColor*, Side *side*)

Is enabled boolean.

Parameters

playerColor	the player color
side	the side

Returns

the boolean

static void main (String[] *args*) `[static]`

The entry point of application.

Parameters

args	the input arguments

String toString ()

To string string.

Returns

the string

static PlayerColor whosCastling (byte *castlingRights*) `[static]`

Whos castling player color.

Parameters

castlingRights	the castling rights

Returns

the player color

Member Data Documentation

final String NO_CASTLING_ABILITY `[static]`

The constant NO_CASTLING_ABILITY.

MyJFrame.Closing< T >

Public Attributes

- String **title**
- String **header**
- ImageIcon **icon**

Detailed Description

The interface **Closing**.

Parameters

<T>	the type parameter

Member Function Documentation

boolean checkClosingVal (T *val*)

Check closing val boolean.

Parameters

val	the val

Returns

the boolean

void closing (T *val*)

Closing.

Parameters

val	the val

T show ()

Show t.

Returns

the t

default void tryClose ()

Try close.

Member Data Documentation

String header

The constant header.

ImageIcon icon

The constant icon.

String title

The constant title.

Col

Static Public Attributes

- static final **Col GameID**
- static final **Col SavedGame**
- static final **Col Password**
- static final **Col Player1**
- static final **Col Player2**
- static final **Col Username**
- static final **Col Winner**
- static final **Col PlayerToMove**
- static final **Col CreatedDateTime**
- static final **Col ProfilePic**

Detailed Description

Col - represents a column. either existing column in the db (the constant columns **`GameID`** , **`SavedGame`** ...) or created columns.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Col (Col *col*)

Instantiates a new **Col**.

Parameters

col	the col

Col (String *colName*, String *alias*)

Instantiates a new **Col**.

Parameters

colName	the col name
alias	the alias

Col (String *colName*)

Instantiates a new **Col**.

Parameters

colName	the col name

Member Function Documentation

Col as ()

As own name. practically keeps the name's case.

Returns

the col

Col as (String *alias*)

creates a new column with the given alias as its alias

Parameters

alias	the alias

Returns

the col

String colName ()

Col name string.

Returns

the string

static Col count (String *as*)`[static]`

Count col. counts every row (*)

Parameters

as	the alias

Returns

the col

static Col count (String *as*, Object *countWhat*)`[static]`

Count col.

Parameters

as	the alias
countWhat	the count what

Returns

the col

static Col countIf (String *as*, Condition *condition*)`[static]`

Count if col. only counts if the given condition is true

Parameters

as	the as
condition	the condition

Returns

the col

Col date ()

Date col.

Returns

a new col representing datetime

String label ()

Label string.

Returns

the string

Col math (Math *operation*, Object *value*)

Math col.

Parameters

operation	the operation
value	the value

Returns

the col

Col math (Math *operation*, Object *value*, boolean *changeSelf*)

Math col.

Parameters

operation	the operation
value	the value

changeSelf	the change self

Returns

the col

String nested ()

Nested string.

Returns

the string

Col of (String *ofWhom*)

new col like this that belongs to the given owner

Parameters

ofWhom	the of whom

Returns

the col

Col of (Table *table*)

new col like this that belongs to the given table

Parameters

table	the table

Returns

the col

Col replace (String *replacing*, String *replaceWith*)

Replace col.

Parameters

replacing	the replacing
replaceWith	the replace with

Returns

the col

void setColName (String *colName*)

Sets col name.

Parameters

colName	the col name

void setWrapped (boolean *wrapped*)

Sets wrapped.

Parameters

wrapped	the wrapped

static CustomCol sum (String *as*, Col... *colsToSum*)`[static]`

Sum cols.

Parameters

as	the alias
colsToSum	the cols to sum

Returns

the custom col

static Col switchCase (String *as*, SwitchCase... *cases*) `[static]`

Switch case col.

Parameters

as	the as
cases	the cases

Returns

the col

Col time ()

Time col.

Returns

a new col representing time

String toString ()

To string string.

Returns

the string

void wrap ()

Wrap.

Member Data Documentation

final Col CreatedDateTime `[static]`

The constant CreatedDateTime.

final Col GameID `[static]`

The constant GameID.

final Col Password `[static]`

The constant Password.

final Col Player1 `[static]`

The constant Player1.

final Col Player2 `[static]`

The constant Player2.

final Col PlayerToMove `[static]`

The constant PlayerToMove.

final Col ProfilePic `[static]`

The constant ProfilePic.

final Col SavedGame `[static]`

The constant SavedGame.

final Col Username `[static]`

The constant Username.

final Col Winner `[static]`

The constant Winner.

Condition

Classes

- enum **Relation**

Detailed Description

Condition - represents a condition.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Class Documentation

enum ver14::SharedClasses::DBActions::Condition::Relation

Relation - relations between conditions.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Enumerator:

AND	And relation.
OR	Or relation.

Constructor & Destructor Documentation

Condition (String *str*, Object... *parms*)

Instantiates a new **Condition**.

Parameters

str	the str
parms	the parms

Member Function Documentation

Condition add (Condition *condition*, Relation *relation*)

Add condition.

Parameters

condition	the condition
relation	the relation

Returns

the condition

Condition add (Condition *condition*, Relation *relation*, boolean *wrap*)

Add condition.

Parameters

condition	the condition
relation	the relation
wrap	the wrap

Returns

THIS condition

Condition and (Condition *condition*)

wraps

Parameters

condition	the condition

Returns

condition

static Condition between (Object *col*, Object *start*, Object *end*)`[static]`

Between condition.

Parameters

col	the col
start	the start
end	the end

Returns

the condition

static Condition equals (Object *col*, Object *value*)`[static]`

Equals condition.

Parameters

col	the col
value	the value

Returns

the condition

String getStr ()

Gets str.

Returns

the str

static Condition math (Object *col*, @MagicConstant(stringValues={">", ">=", "<", "<="}) String *operation*, Object *value*)`[static]`

Math condition.

Parameters

col	the col
operation	the operation
value	the value

Returns

the condition

Condition noNulls ()

No nulls condition.

Returns

the condition

static Condition notEquals (Object *col*, Object *value*)`[static]`

Not equals condition.

Parameters

col	the col
value	the value

Returns

the condition

void setStr (String *str*)

Sets str.

Parameters

str	the str

String toString ()

To string string.

Returns

the string

void wrap ()

Wrap.

Config< V >

Public Attributes

- final boolean **canUseDefault**
- final String **description**

Detailed Description

Config - an argument's configuration.

Parameters

<V>	the type

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Config ()

Instantiates a new **Config**.

Config (String *description*)

Instantiates a new **Config**.

Parameters

description	the description

Config (String *description*, boolean *canUseDefault*, Described< V > *defaultValue*)

Instantiates a new **Config**.

Parameters

description	the description
canUseDefault	the can use default
defaultValue	the default value

Config (String *description*, V *defVal*)

Instantiates a new **Config**.

Parameters

description	the description
defVal	the def val

Config (String *description*, V *defVal*, String *defDesc*)

Instantiates a new **Config**.

Parameters

description	the description
defVal	the def val
defDesc	the def desc

Config (String *description*, Described< V > *defaultValue*)

Instantiates a new **Config**.

Parameters

description	the description
defaultValue	the default value

Member Function Documentation

void addSuggestion (Described< V > *suggestion*)

Add suggestion.

Parameters

suggestion	the suggestion

V getDefault ()

Gets default.

Returns

the default

String getDefaultDesc ()

Gets default desc.

Returns

the default desc

Described< V > getDescribedDefault ()

Gets described default.

Returns

the described default

ArrayList< Described< V > > getValuesSuggestion ()

Gets values suggestion.

Returns

the values suggestion

String toString ()

To string string.

Returns

the string

Member Data Documentation

final boolean canUseDefault

The Can use default.

final String description

The Description.

ConfirmDialogs

Detailed Description

The utility class Confirm dialogs.

Member Function Documentation

static String confirm (Component *parent*, String *title*, String *header*, String *message*, ImageIcon *icon*, String *initialValue*) `[static]`

Confirm string.

Parameters

parent	the parent
header	the header
message	the message
title	the title
icon	the icon
initialValue	the initial value

Returns

the string

static boolean confirm (Component *parent*, String *title*, String *message*, ImageIcon *icon*) `[static]`

Confirm boolean.

Parameters

parent	the parent
title	the title
message	the message
icon	the icon

Returns

the boolean

static void main (String[] *args*) `[static]`

The entry point of application.

Parameters

args	the input arguments

CreatedGame

Additional Inherited Members

Detailed Description

Created game.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

CreatedGame (String *gameId*, String *creatorUsername*, GameSettings *gameSettings*)

Instantiates a new Created game.

Parameters

gameId	the game id
creatorUsername	the creator username
gameSettings	the game settings

Member Function Documentation

String getGameDesc ()

Gets game desc.

Returns

the game desc

Reimplemented from **GameInfo** (*p.47*).

CustomStatement

Additional Inherited Members

Detailed Description

Custom statement - a custom sql statement.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

CustomStatement (DBRequest.Type *type*, @Language("SQL") String *statement*)

Instantiates a new Custom statement.

Parameters

type	the type
statement	the statement

Member Function Documentation

String createStatement () `[protected]`

Create statement string.

Returns

the string

Reimplemented from **SQLStatement** (*p.120*).

DBRequest

Classes

- enum **Type**

Public Attributes

- final **Type type**

Detailed Description

Db request - a database request.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Class Documentation

enum ver14::SharedClasses::DBActions::DBRequest::DBRequest::Type

Type - .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Enumerator:

Query	Query type.
Update	Update type.

Constructor & Destructor Documentation

DBRequest (SQLStatement *sqlStatement*)

Instantiates a new Db request.

Parameters

sqlStatement	the sql statement

DBRequest (Type *type*, String *request*, RequestBuilder *builder*)

Instantiates a new Db request.

Parameters

type	the type
request	the request
builder	the builder

DBRequest (SQLStatement *sqlStatement*, RequestBuilder *builder*)

Instantiates a new Db request.

Parameters

sqlStatement	the sql statement
builder	the builder

Member Function Documentation

RequestBuilder getBuilder ()

Gets builder.

Returns

the builder

String getRequest ()

Gets request.

Returns

the request

DBRequest getSubRequest ()

Gets sub request.

Returns

the sub request

void setSubRequest (DBRequest *subRequest*)

Sets sub request.

Parameters

subRequest	the sub request

String toString ()

To string string.

Returns

the string

Member Data Documentation

final Type type

The **Type**.

DBResponse

Classes

- enum **Status**

Protected Attributes

- final **Status status**
- final **DBRequest request**
- **DBResponse addedRes**

Detailed Description

Db response - .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Class Documentation

enum ver14::SharedClasses::DBActions::DBResponse::DBResponse::Status

Status - .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Enumerator:

ERROR	Error status.
SUCCESS	Success status.

Constructor & Destructor Documentation

DBResponse (Status *status*, DBRequest *request*) `[protected]`

Instantiates a new Db response.

Parameters

status	the status
request	the request

Member Function Documentation

abstract DBResponse clean () `[abstract]`

Clean db response.

Returns

the db response

Reimplemented in **StatusResponse** (*p.123*), and **TableDBResponse** (*p.133*).

DBResponse getAddedRes ()

Gets added res.

Returns

the added res

DBRequest getRequest ()

Gets request.

Returns

the request

Status getStatus ()

Gets status.

Returns

the status

abstract boolean isAnyData () `[abstract]`

Is any data boolean.

Returns

the boolean

Reimplemented in **StatusResponse** (*p.124*), and **TableDBResponse** (*p.134*).

boolean isSuccess ()

Is success boolean.

Returns

the boolean

void print ()

Print.

void setAddedRes (DBResponse *addedRes*)

Sets added res.

Parameters

addedRes	the added res

String toString ()

To string string.

Returns

the string

Reimplemented in **TableDBResponse** (*p.134*).

Member Data Documentation

DBResponse addedRes `[protected]`

The Added res.

final DBRequest request `[protected]`

The Request.

final Status status `[protected]`

The **Status**.

Delete

Additional Inherited Members

Detailed Description

Delete - deletion statement.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Delete (Table *deletingFrom*, Condition *condition*)

Instantiates a new **Delete**.

Parameters

deletingFrom	the deleting from
condition	the condition

Member Function Documentation

String createStatement () `[protected]`

Create statement string.

Returns

the string

Reimplemented from **SQLStatement** (*p.120*).

Direction Enum Reference

Public Attributes

- **U**
- **D**
- **L**
- **R**
- **U_U**
- **D_D**
- **U_R**
- **U_L**
- **D_R**
- **D_L**
- **U_U_R**
- **U_U_L**
- **U_R_R**
- **U_L_L**
- **D_D_R**
- **D_D_L**
- **D_R_R**
- **D_L_L**
- final long **andWith**
- final int **offset**
- final int **asInt**
- final **Direction**[] **combination**

Static Public Attributes

- static final int **NUM_OF_DIRECTIONS**
- static final int **NUM_OF_KNIGHT_DIRECTIONS**
- static final int **NUM_OF_DIRECTIONS_WO_KNIGHT**
- static final **Direction**[] **ALL_DIRECTIONS**
- static final List< **Direction** > **ALL_USED_DIRECTIONS**
- static final **PlayerColor normalPerspective**

Detailed Description

Direction - represents a moving direction on a board. sort of like a vector. has an **`offset`** that is added to a certain location or bitboard, in order to achieve movement in that direction. the general direction map looks like this:

-9	-8	-7
-1	loc	1
7	8	9

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

Direction[] getCombination ()

Get combination direction [ ].

Returns

the direction [ ]

static Direction getDirectionByOffset (int *offset*) `[static]`

Gets direction by offset.

Parameters

offset	the offset

Returns

the direction by offset

static Direction getRelative (Location *loc1*, Location *loc2*) `[static]`

Gets relative.

Parameters

loc1	the loc 1
loc2	the loc 2

Returns

the relative

abstract Direction opposite () `[abstract]`

the Opposite direction to this one.

Returns

the direction

Direction perspective (PlayerColor *playerColor*)

gets the correct perspective for the provided player color.this is necessary because for example: a white pawn push(`U`) is the exact opposite of a black pawn push (`D`). so the perspective needs to be in relation to the moving color.

Parameters

playerColor	the player color

Returns

the direction

Member Data Documentation

final Direction [] ALL_DIRECTIONS `[static]`

The All directions.

final List<Direction> ALL_USED_DIRECTIONS `[static]`

The All used directions.

final long andWith

some directions need to filter false positives. for example: moving left one square from the left-most column, will overflow to the previous row. to fix this problem some directions have a andWith value they have to perform a bitwise and with, after every offset. to cancel the false positives. in the left direction example, the andWith is the whole board but the right-most column

final int asInt

an int for quick access by index.

final Direction [] combination

some

D

one square down the board.

D_D

two squares down.

D_D_L

two squares down and one square left.

D_D_R

two squares down and one square right.

D_L

one square down and one square left.

D_L_L

two squares left and one square down.

D_R

one square down and one square right.

D_R_R

two squares right and one square down.

L

one square left.

final PlayerColor normalPerspective `[static]`

the perspective the offset is correct for. if the moving piece's color is not this value the direction need to be flipped.

final int NUM_OF_DIRECTIONS `[static]`

The constant NUM_OF_DIRECTIONS.

final int NUM_OF_DIRECTIONS_WO_KNIGHT `[static]`

The constant NUM_OF_DIRECTIONS_WO_KNIGHT.

final int NUM_OF_KNIGHT_DIRECTIONS `[static]`

The constant NUM_OF_KNIGHT_DIRECTIONS.

final int offset

The actual offset.

R

one square right.

U

one square up the board.

U_L

one square up and one square left.

U_L_L

two squares left and one square up.

U_R

one square up and one square right.

U_R_R

two squares right and one square up.

U_U

two squares up.

U_U_L

two squares up and one square left.

U_U_R

two squares up and one square right.

EnvManager

Detailed Description

Env manager - an object that can handle errors as they occur.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

void criticalErr (MyError *err*)

notifies manager of an un-handleable error. the manager must shut down everything

Parameters

err	the error thrown

void handledErr (MyError *err*)

notifies manager of a managed error

Parameters

err	the error thrown

ErrorHandler< E extends MyError >

Detailed Description

The interface Error handler.

Parameters

<E>	the type parameter

Member Function Documentation

void handle (MyError *err*)

Handle.

Parameters

err	the err

static boolean ignore (ThrowingRunnable *runnable*)`[static]`

Ignore boolean.

Parameters

runnable	the runnable

Returns

true if the runnable threw, false otherwise

EstablishedGameInfo

Public Attributes

- final String **opponentUsername**

Protected Attributes

- Date **createdAt**

Detailed Description

Established game info.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

EstablishedGameInfo (String *gameId*, String *creatorUsername*, String *opponentUsername*, GameSettings *gameSettings*, Stack< Move > *moveStack*) `[protected]`

Instantiates a new Established game info.

Parameters

gameId	the game id
creatorUsername	the creator username
opponentUsername	the opponent username
gameSettings	the game settings
moveStack	the move stack

Member Function Documentation

Date getCreatedAt ()

Gets created at.

Returns

the created at

String getGameDesc ()

Gets game desc.

Returns

the game desc

Reimplemented from **GameInfo** (*p.47*).

Stack< Move > getMoveStack ()

Gets move stack.

Returns

the move stack

void setCreatedAt (Date *createdAt*)

Sets created at.

Parameters

createdAt	the created at

String toString ()

To string string.

Returns

the string

Reimplemented from **GameInfo** (*p.48*).

Reimplemented in **ArchivedGameInfo** (*p.4*).

Member Data Documentation

Date createdAt`[protected]`

The Created at.

final String opponentUsername

The Opponent username.

Evaluation

Static Public Attributes

- static final int **TIE_EVAL**
- static final int **WIN_EVAL**
- static final int **LOSS_EVAL**

Detailed Description

Evaluation.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Evaluation (GameStatus *gameStatus*, PlayerColor *evaluationFor*)

Instantiates a new **Evaluation**.

Parameters

gameStatus	the game status
evaluationFor	the evaluation for

Evaluation (int *eval*, GameStatus *gameStatus*, PlayerColor *evaluationFor*)

Instantiates a new **Evaluation**.

Parameters

eval	the eval
gameStatus	the game status
evaluationFor	the evaluation for

Evaluation (PlayerColor *evaluationFor*)

Instantiates a new **Evaluation**.

Parameters

evaluationFor	the evaluation for

Evaluation (Evaluation *other*)

Instantiates a new **Evaluation**.

Parameters

other	the other

Member Function Documentation

void addDetail (EvaluationParameters *parm*, int *value*)

Add detail.

Parameters

parm	the parm
value	the value

void assertNotGameOver ()

Assert not game over.

static Evaluation book () `[static]`

Book evaluation.

Returns

the evaluation

float convertFromCentipawns ()

Convert from centipawns float.

Returns

the float

boolean equals (Object *o*)

Equals boolean.

Parameters

o	the o

Returns

the boolean

record EvaluationDetail (EvaluationParameters *parm*, double *eval*)

Evaluation detail.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

void flipEval ()

Flip eval.

int getEval ()

Gets eval.

Returns

the eval

Integer getEvaluationDepth ()

Gets evaluation depth.

Returns

the evaluation depth

PlayerColor getEvaluationFor ()

Gets evaluation for.

Returns

the evaluation for

GameStatus getGameStatus ()

Gets game status.

Returns

the game status

boolean isCheck ()

Is check boolean.

Returns

the boolean

boolean isGameOver ()

Is game over boolean.

Returns

the boolean

boolean isGreaterThan (Evaluation *other*)

Is greater than boolean.

Parameters

other	the other

Returns

the boolean

void print ()

Print.

void setEval (int *eval*)

Sets eval.

Parameters

eval	the eval

void setEvaluationDepth (Integer *evaluationDepth*)

Sets evaluation depth.

Parameters

evaluationDepth	the evaluation depth

Evaluation setPerspective (PlayerColor *playerColor*)

Sets perspective.

Parameters

playerColor	the player color

Returns

the perspective

String toString ()

To string string.

Returns

the string

Member Data Documentation

final int LOSS_EVAL [static]

The constant LOSS_EVAL.

final int TIE_EVAL [static]

The constant TIE_EVAL.

final int WIN_EVAL [static]

The constant WIN_EVAL.

EvaluationParameters Enum Reference

Public Attributes

- **MATERIAL**
- **PIECE_TABLES**
- **KING_SAFETY**
- **HANGING_PIECES**
- **SQUARE_CONTROL**
- **MOVEMENT_ABILITY**
- **FORCE_KING_TO_CORNER**
- **EG_WEIGHT**
- **STOCKFISH_SAYS**
- final double **weight**

Detailed Description

Evaluation parameters - all evaluation parameters. (some are unused).

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Data Documentation

EG_WEIGHT

Eg weight evaluation parameters.

FORCE_KING_TO_CORNER

Force king to corner evaluation parameters.

HANGING_PIECES

Hanging pieces evaluation parameters.

KING_SAFETY

King safety evaluation parameters.

MATERIAL

Material evaluation parameters.

MOVEMENT_ABILITY

Movement ability evaluation parameters.

PIECE_TABLES

Piece tables evaluation parameters.

SQUARE_CONTROL

Square control evaluation parameters.

STOCKFISH_SAYS

Stockfish says evaluation parameters.

final double weight

The parameter's weight

GameInfo

Public Attributes

- final String **gameId**
- final String **creatorUsername**
- final GameSettings **gameSettings**

Detailed Description

Game info.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

GameInfo (String *gameId*, String *creatorUsername*, GameSettings *gameSettings*) `[protected]`

Instantiates a new Game info.

Parameters

gameId	the game id
creatorUsername	the creator username
gameSettings	the game settings

Member Function Documentation

boolean equals (Object *o*)

Equals boolean.

Parameters

o	the o

Returns

the boolean

static GameInfo example () `[static]`

Example game info.

Returns

the game info

abstract String getGameDesc () `[abstract]`

Gets game desc.

Returns

the game desc

Reimplemented in **CreatedGame** (*p.30*), and **EstablishedGameInfo** (*p.41*).

PlayerColor getJoiningPlayerColor ()

Gets joining player color.

Returns

the joining player color

PlayerColor getStartingColor ()

Gets starting color.

Returns

the starting color

String ID ()

Id string.

Returns

the string

boolean isCreator (String *username*)

Is creator boolean.

Parameters

username	the username

Returns

the boolean

String toString ()

To string string.

Returns

the string

Reimplemented in **ArchivedGameInfo** (*p.4*), and **EstablishedGameInfo** (*p.41*).

Member Data Documentation

final String creatorUsername

The Creator username.

final String gameId

The Game id.

final GameSettings gameSettings

The Game settings.

GameStatus

Classes

- enum **GameStatusType**
- enum **SpecificStatus**

Detailed Description

Game status - represents a game status .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

static GameStatus checkmate (PlayerColor *winningPlayerColor*, Location *matedKing*) `[static]`

Checkmate game status.

Parameters

winningPlayerColor	the winning player color
matedKing	the mated king

Returns

the game status

static GameStatus fiftyMoveRule () `[static]`

Fifty move rule game status.

Returns

the game status

static GameStatus gameGoesOn () `[static]`

Game goes on game status.

Returns

the game status

Location getCheckedKingLoc ()

Gets checked king loc.

Returns

the checked king loc

String getDetailedStr ()

Gets detailed str.

Returns

the detailed str

String getDetailedStr (Map< PlayerColor, String > *playerUsernamesMap*)

Gets detailed str.

Parameters

playerUsernamesMap	the player usernames map

Returns

the detailed str

GameStatusType getGameStatusType ()

Gets game status type.

Returns

the game status type

SpecificStatus getSpecificStatus ()

Gets specific status.

Returns

the specific status

PlayerColor getWinningColor ()

Gets winning color.

Returns

the winning color

static GameStatus insufficientMaterial () `[static]`

Insufficient material game status.

Returns

the game status

boolean isCheck ()

Is check boolean.

Returns

the boolean

boolean isDisconnected ()

Is disconnected boolean.

Returns

the boolean

boolean isGameOver ()

Is game over boolean.

Returns

the boolean

static GameStatus playerDisconnected (PlayerColor *disconnectedPlayer*, boolean *isVsAi*) `[static]`

Player disconnected game status.

Parameters

disconnectedPlayer	the disconnected player
isVsAi	the is vs ai

Returns

the game status

static GameStatus playerResigned (PlayerColor *resignedPlayer*) `[static]`

Player resigned game status.

Parameters

resignedPlayer	the resigned player

Returns

the game status

static GameStatus serverStoppedGame (String *cause*)`[static]`

Server stopped game game status.

Parameters

cause	the cause

Returns

the game status

void setCustomStr (String *customStr*)

Sets custom str.

Parameters

customStr	the custom str

void setDepth (int *depth*)

Sets depth.

Parameters

depth	the depth

void setInCheck (Location *checkedKingLoc*)

Sets in check.

Parameters

checkedKingLoc	the checked king loc

static GameStatus stalemate ()`[static]`

Stalemate game status.

Returns

the game status

static GameStatus threeFoldRepetition ()`[static]`

Three fold repetition game status.

Returns

the game status

static GameStatus tieByAgreement ()`[static]`

Tie by agreement game status.

Returns

the game status

static GameStatus timedOut (PlayerColor *timedOutPlayer*, boolean *isSufficientMaterial*)`[static]`

Timed out game status.

Parameters

timedOutPlayer	the timed out player
isSufficientMateria	the is sufficient material

l	

Returns

the game status

String toString ()

To string string.

Returns

the string

GameStatus.GameStatusType Enum Reference

Public Attributes

- **TIE**
- **CHECK**
- **GAME_GOES_ON**
- **WIN_OR_LOSS**
- **UNFINISHED**
- final String **annotation**
- final String **gameOverStr**

Detailed Description

Game status type .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

GameStatusType (String *annotation*)

Instantiates a new Game status type.

Parameters

annotation	the annotation

GameStatusType (String *annotation*, String *gameOverStr*)

Instantiates a new Game status type.

Parameters

annotation	the annotation
gameOverStr	the game over str

Member Function Documentation

boolean isGameOver ()

Is game over.

Returns

true if is game over. false otherwise

Member Data Documentation

final String annotation

The game status annotation.

CHECK

Check game status type.

GAME_GOES_ON

Game goes on game status type.

final String gameOverStr

game over str

TIE

Tie game status type.

UNFINISHED

Unfinished game status type.

WIN_OR_LOSS

Win or loss game status type.

HandledThread

Detailed Description

The type Handled thread.

Constructor & Destructor Documentation

HandledThread ()

Instantiates a new Handled thread.

HandledThread (ThrowingRunnable *runnable*)

Instantiates a new Handled thread.

Parameters

runnable	the runnable

Member Function Documentation

void handledRun () throws Throwable [protected]

Handled run.

Exceptions

Throwable	the throwable

Reimplemented from **MyThread** (*p.95*).

static HandledThread runInHandledThread (ThrowingRunnable *runnable*) [static]

Run in handled thread handled thread.

Parameters

runnable	the runnable

Returns

the handled thread

void setRunnable (ThrowingRunnable *runnable*)

Sets runnable.

Parameters

runnable	the runnable

MyJFrame.MyAdapter.HeldDown

Detailed Description

The interface Held down.

Member Function Documentation

void endPress ()

End press.

int key ()

Key int.

Returns

the int

void startPress ()

Start press.

IDsGenerator

Detailed Description

IDs generator.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

boolean canUseId (String *id*)

Can use id boolean.

Parameters

id	the id

Returns

the boolean

synchronized String generate ()

Generate id.

Returns

the string

StrUtils.IterationThingy< T >

Detailed Description

Iteration thingy.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

T iteration (int *i*, boolean *isLast*)

Iteration t.

Parameters

i	the
isLast	the is last

Returns

the t

Location Enum Reference

Public Attributes

- final long **asLong**
- final int **asInt**
- final int **row**
- final int **col**

Detailed Description

Location - an enum consisting of 64 values representing all 64 squares on the board. used to access squares on the board

an enum is used for performance reasons.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Location ()

Instantiates a new **Location**.

Member Function Documentation

static String createMatIndicesStr () `[static]`

Create mat indices str string.

Returns

the string

Location flip ()

Flip location.

Returns

the location

static int flip (int *num*) `[static]`

Flip int.

Parameters

num	the num

Returns

the int

String getColString ()

Gets col string.

Returns

the col string

static Location getLoc (int *locIndex*) `[static]`

Gets location corresponding to the locIndex provided (0..63)

Parameters

locIndex	the locIndex

Returns

the location if the provided index is inside the bounds(0...63). null otherwise

static Location getLoc (int *row*, int *col*) `[static]`

Gets loc.

Parameters

row	the row
col	the col

Returns

the loc

static Location getLoc (int *row*, int *col*, boolean *flip*) `[static]`

Gets loc.

Parameters

row	the row
col	the col
flip	the flip

Returns

the loc

static Location getLoc (Location *loc*, Direction *direction*) `[static]`

Gets the location relative to loc in the direction

Parameters

loc	the loc
direction	the direction

Returns

the location if the calculated index is inside the bounds(0...63). null otherwise

static Location getLoc (Location *loc*, int *add*) `[static]`

Gets the location that is exactly add squares from loc NOTE: add should be in bitboard format

Parameters

loc	the loc
add	the number of squares to add

Returns

the location if the calculated index is inside the bounds(0...63). null otherwise

See also

ver14.Model.Bitboard

static Location getLoc (Location *loc*, int *numOfMult*, Direction *direction*)`[static]`

Gets the location relative to loc in the direction given and the distance is determined by the numOfMult

Parameters

loc	the loc
numOfMult	the num of mult
direction	the direction

Returns

the location if the calculated index is inside the bounds(0...63). null otherwise

static Location getLoc (String *str*)`[static]`

gets the location corresponding to the provided location string.

Parameters

str	the square's coordinate according to the Algebraic notation

Returns

the loc if given str is valid. null otherwise

See also

. . .

int getMaxDistance (Location *other*)

Gets max distance.

Parameters

other	the other

Returns

the max distance

String getRowString ()

Gets row string.

Returns

the row string

int hash (PieceType *pieceType*)

Hash int.

Parameters

pieceType	the piece type

Returns

the int

boolean isBlackSquare ()

Is black square boolean.

Returns

the boolean

static boolean isInBounds (Location *loc*) `[static]`

Is in bounds boolean.

Parameters

loc	the loc

Returns

the boolean

boolean isWhiteSquare ()

Is white square boolean.

Returns

the boolean

static Location[][] locsMat () `[static]`

Locs mat location [ ] [ ].

Returns

the location [ ] [ ]

static String locsMatStr () `[static]`

Locs mat str string.

Returns

the string

String matrixStr ()

Matrix str string.

Returns

the string

String toString ()

To string string.

Returns

the string

static Location valueOf (int *locIndex*) `[static]`

Value of location.

Parameters

locIndex	the locIndex

Returns

the location if the provided index is inside the bounds(0...63). null otherwise

Member Data Documentation

final int asInt

the calculated index

final long asLong

a long value with a bit set on this location

final int col

this location's column

final int row

this location's row

LoginInfo

Detailed Description

Login info.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

LoginInfo ()

Instantiates a new Login info.

LoginInfo (String *username*, String *password*, LoginType *loginType*)

Instantiates a new Login info.

Parameters

username	the username
password	the password
loginType	the login type

LoginInfo (LoginType *loginType*)

Instantiates a new Login info.

Parameters

loginType	the login type

Member Function Documentation

boolean asUser ()

As user boolean.

Returns

the boolean

LoginType getLoginType ()

Gets login type.

Returns

the login type

String getPassword ()

Gets password.

Returns

the password

String getProfilePic ()

Gets profile pic.

Returns

the profile pic

String getUsername ()

Gets username.

Returns

the username

void initDebugLoginValues ()

Init debug login values.

boolean isGuest ()

Is guest boolean.

Returns

the boolean

void setLoginType (LoginType *loginType*)

Sets login type.

Parameters

loginType	the login type

void setPassword (String *password*)

Sets password.

Parameters

password	the password

void setProfilePic (String *profilePic*)

Sets profile pic.

Parameters

profilePic	the profile pic

void setUsername (String *username*)

Sets username.

Parameters

username	the username

String toString ()

To string string.

Returns

the string

LoginType Enum Reference

Public Attributes

- **LOGIN**
- **REGISTER**
- **GUEST**
- **CANCEL**
- **NOT_SET_YET**

Detailed Description

Login Type.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

boolean asUser ()

As user boolean.

Returns

the boolean

String toString ()

To string string.

Returns

the string

Member Data Documentation

CANCEL

Cancel.

GUEST

Guest.

LOGIN

Login.

NOT_SET_YET

Not set yet.

REGISTER

Register.

Math Enum Reference

Public Attributes

- **Plus**
- **Mult**
- **Div**
- **Col col**

Detailed Description

Math - allows for math actions on columns and some math-related utilities for columns.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

abstract void apply (Object *value*)`[abstract]`, `[protected]`

Apply.

Parameters

value	the value

static String asFloat (Object *num*)`[static]`

As float string.

Parameters

num	the num

Returns

the string

Col execute (Col *col*, Object *value*)

Execute col.

Parameters

col	the col
value	the value

Returns

the col

Col execute (Col *col*, Object *value*, boolean *changeSelf*)

Execute col.

Parameters

col	the col
value	the value
changeSelf	the change self

Returns

the col

static String formatNum (Object *num*)`[static]`

Format num string.

Parameters

num	the num

Returns

the string

static String formatNum (Object *num*, String *format*)`[static]`

Format num string.

Parameters

num	the num
format	the format

Returns

the string

static String nullIf0 (Object *val*)`[static]`

Null if 0 string.

Parameters

val	the val

Returns

the string

void simpleOperation (Object *a*, String *operation*, Object *b*)`[protected]`

Simple operation.

Parameters

a	the a
operation	the operation
b	the b

static String str (Object *obj*)`[static]`

Str string.

Parameters

obj	the obj

Returns

the string

String strSource ()

Str source string.

Returns

the string

static String strVal (Object *val*) `[static]`

Str val string.

Parameters

val	the val

Returns

the string

void zeroIfNull () `[protected]`

Zero if null.

Member Data Documentation

Col col

The **Col**.

Div

The Div.

Mult

The Mult.

Plus

The Plus.

Message

Public Attributes

- final String **messageID**

Detailed Description

Message.

by Ilan Peretz (`ilanperets@gmail.com`) 10/11/2021

Constructor & Destructor Documentation

Message (MessageType *messageType*)

Instantiates a new **Message**.

Parameters

messageType	the message type

Message (MessageType *messageType*, String *subject*, String *respondingToMsgId*)

Instantiates a new **Message**.

Parameters

messageType	the message type
subject	the subject
respondingToMsgId	the responding to msg id

Message (MessageType *messageType*, Message *respondingTo*)

Instantiates a new **Message**.

Parameters

messageType	the message type
respondingTo	the responding to

Message (MessageType *messageType*, String *subject*)

Instantiates a new **Message**.

Parameters

messageType	the message type
subject	the subject

Member Function Documentation

static Message answerQuestion (Question.Answer *answer*, Message *respondingTo*) `[static]`

Answer question message.

Parameters

respondingTo	the responding to

Returns

the message

static Message askForGameSettings (SyncedItems *joinableGames*, SyncedItems *resumableGames*) `[static]`

Ask for game settings message.

Parameters

joinableGames	the joinable games
resumableGames	the resumable games

Returns

the message

static Message askForLogin () `[static]`

Ask for login message.

Returns

the message

static Message askQuestion (Question *question*) `[static]`

Ask question message.

Parameters

question	the question

Returns

the message

static Message bye (String *subject*) `[static]`

Bye message.

Parameters

subject	the subject

Returns

the message

static Message checkUsernameAvailability (String *username*) `[static]`

Check username availability message.

Parameters

username	the username

Returns

the message

static Message dbRequest (DBRequest *dbRequest*) `[static]`

Db request message.

Parameters

dbRequest	the db request

Returns

the message

static Message error (String *err*) `[static]`

Error message.

Parameters

err	the err

Returns

the message

static Message gameOver (GameStatus *gameStatus*) `[static]`

Game over message.

Parameters

gameStatus	the game status

Returns

the message

Boolean getAvailable ()

Gets available.

Returns

the available

Board getBoard ()

Gets board.

Returns

the board

DBRequest getDBRequest ()

Gets db request.

Returns

the db request

DBResponse getDBResponse ()

Gets db response.

Returns

the db response

MyError getError ()

Gets error.

Returns

the error

GameSettings getGameSettings ()

Gets game settings.

Returns

the game settings

GameStatus getGameStatus ()

Gets game status.

Returns

the game status

GameTime getGameTime ()

Gets game time.

Returns

the game time

LoginInfo getLoginInfo ()

Gets login info.

Returns

the login info

MessageType getMessageType ()

Gets message type.

Returns

the message type

Move getMove ()

Gets move.

Returns

the move

Stack< Move > getMoveStack ()

Gets move stack.

Returns

the move stack

String getOtherPlayer ()

Gets other player.

Returns

the other player

PlayerColor getPlayerColor ()

Gets player color.

Returns

the player color

MovesList getPossibleMoves ()

Gets possible moves.

Returns

the possible moves

ArrayList< Move > getPreMoves ()

Gets pre moves.

Returns

the pre moves

Question getQuestion ()

Gets question.

Returns

the question

String getRespondingToMsgId ()

Gets responding to msg id.

Returns

the responding to msg id

String getSubject ()

Gets subject.

Returns

the subject

SyncedItems<?>[] getSyncedLists ()

Get synced lists synced items [ ].

Returns

the synced items [ ]

String getUsername ()

Gets username.

Returns

the username

ArrayList< String > getUsernameSuggestions ()

Gets username suggestions.

Returns

the username suggestions

static Message initGame (Board *board*, String *opponent*, PlayerColor *player*, GameTime *gameTime*, Stack< Move > *moveStack*) `[static]`

Init game message.

Parameters

board	the board
opponent	the opponent
player	the player
gameTime	the game time
moveStack	the move stack

Returns

the message

static Message interrupt () `[static]`

Interrupt message.

Returns

the message

boolean isResponse ()

Is response boolean.

Returns

the boolean

boolean isSubject ()

Is subject boolean.

Returns

the boolean

static Message returnDBResponse (DBResponse *DBResponse*, Message *request*) `[static]`

Return db response message.

Parameters

DBResponse	the db response
request	the request

Returns

the message

static Message returnGameSettings (GameSettings *gameSettings*, Message *respondingTo*) `[static]`

Return game settings message.

Parameters

gameSettings	the game settings
respondingTo	the responding to

Returns

the message

static Message returnLogin (LoginInfo *loginInfo*, Message *respondingTo*) `[static]`

Return login message.

Parameters

loginInfo	the login info
respondingTo	the responding to

Returns

the message

static Message returnMove (Move *move*, Message *respondingTo*) `[static]`

Return move message.

Parameters

move	the move
respondingTo	the responding to

Returns

the message

static Message returnUsernameAvailable (Message *request*) `[static]`

Return username available message.

Parameters

request	the request

Returns

the message

static Message returnUsernameNotAvailable (ArrayList< String > *usernameSuggestions*, Message *request*) `[static]`

Return username not available message.

Parameters

usernameSuggestions	the username suggestions
request	the request

Returns

the message

void setAvailable (Boolean *available*)

Sets available.

Parameters

available	the available

void setBoard (Board *board*)

Sets board.

Parameters

board	the board

void setDbRequest (DBRequest *dbRequest*)

Sets db request.

Parameters

dbRequest	the db request

void setDBResponse (DBResponse *requestedStats*)

Sets db response.

Parameters

requestedStats	the requested stats

void setError (MyError *error*)

Sets error.

Parameters

error	the error

void setGameSettings (GameSettings *gameSettings*)

Sets game settings.

Parameters

gameSettings	the game settings

void setGameStatus (GameStatus *gameStatus*)

Sets game status.

Parameters

gameStatus	the game status

void setGameTime (GameTime *gameTime*)

Sets game time.

Parameters

gameTime	the game time

void setLoginInfo (LoginInfo *loginInfo*)

Sets login info.

Parameters

loginInfo	the login info

void setMove (Move *move*)

Sets move.

Parameters

move	the move

void setMoveStack (Stack< Move > *moveStack*)

Sets move stack.

Parameters

moveStack	the move stack

void setOtherPlayer (String *otherPlayer*)

Sets other player.

Parameters

otherPlayer	the other player

void setPlayerColor (PlayerColor *playerColor*)

Sets player color.

Parameters

playerColor	the player color

void setPossibleMoves (MovesList *possibleMoves*)

Sets possible moves.

Parameters

possibleMoves	the possible moves

void setPreMoves (ArrayList< Move > *preMoves*)

Sets pre moves.

Parameters

preMoves	the pre moves

void setQuestion (Question *question*)

Sets question.

Parameters

question	the question

void setRespondingTo (Message *msg*)

Sets responding to.

Parameters

msg	the msg

void setRespondingToMsgId (String *respondingToMsgId*)

Sets responding to msg id.

Parameters

respondingToMsgId	the responding to msg id

void setSyncedLists (SyncedItems<?>... *syncedLists*)

Sets synced lists.

Parameters

syncedLists	the synced lists

void setUsername (String *username*)

Sets username.

Parameters

username	the username

void setUsernameSuggestions (ArrayList< String > *usernameSuggestions*)

Sets username suggestions.

Parameters

usernameSuggestions	the username suggestions

static Message syncLists (SyncedItems<?>... *syncedLists*) `[static]`

Sync lists message.

Parameters

syncedLists	the synced lists

Returns

the message

static Message throwError (MyError *error*) `[static]`

Throw error message.

Parameters

error	if null interrupts

Returns

the message

static Message updateByMove (Move *move*, GameTime *gameTime*) `[static]`

Update by move message.

Parameters

move	the move
gameTime	the game time

Returns

the message

static Message waitForMatch () `[static]`

Wait for match message.

Returns

the message

static Message waitForYourTurn (String *waitingForName*, GameTime *gameTime*)`[static]`

Wait for your turn message.

Parameters

waitingForName	the waiting for name
gameTime	the game time

Returns

the message

static Message welcomeMessage (String *str*, LoginInfo *loginInfo*)`[static]`

Welcome message message.

Parameters

str	the str
loginInfo	the login info

Returns

the message

Member Data Documentation

final String messageID

The **Message** id.

MessageCallback

Detailed Description

Message callback - a message callback.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

void onMsg (Message *message*)

On msg.

Parameters

message	the message

MessagesHandler

Protected Attributes

- final **AppSocket socket**

Detailed Description

The type Messages handler.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

MessagesHandler (AppSocket *socket*)

Instantiates a new Messages handler.

Parameters

socket	the socket

Member Function Documentation

Message blockTilRes (Message *request*)

Block til res message.

Parameters

request	the request

Returns

the message

MyError.DisconnectedError createDisconnectedError () `[protected]`

Create disconnected error my error . disconnected error.

Returns

the my error . disconnected error

void interruptBlocking (MyError *err*)

Interrupt blocking.

Parameters

err	the err

void noBlockRequest (Message *request*, MessageCallback *onRes*)

No block request.

Parameters

request	the request
onRes	the on res

MessageCallback onAddTime ()

On add time message callback.

Returns

the message callback

MessageCallback onAlive ()

On alive message callback.

Returns

the message callback

void onAnyDisconnection () `[protected]`

On any disconnection.

void onAnyMsg (Message *message*)

On any msg.

Parameters

message	the message

MessageCallback onBye ()

On bye message callback.

Returns

the message callback

MessageCallback onCancelQuestion ()

On cancel question message callback.

Returns

the message callback

MessageCallback onDBRequest ()

On db request message callback.

Returns

the message callback

MessageCallback onDBResponse ()

On db response message callback.

Returns

the message callback

final void onDisconnected ()

On disconnected.

MessageCallback onError ()

On error message callback.

Returns

the message callback

MessageCallback onGameOver ()

On game over message callback.

Returns

the message callback

MessageCallback onGetGameSettings ()

On get game settings message callback.

Returns

the message callback

MessageCallback onGetMove ()

On get move message callback.

Returns

the message callback

MessageCallback onInitGame ()

On init game message callback.

Returns

the message callback

MessageCallback onInterrupt ()

On interrupt message callback.

Returns

the message callback

MessageCallback onIsAlive ()

On is alive message callback.

Returns

the message callback

MessageCallback onLogin ()

On login message callback.

Returns

the message callback

MessageCallback onOfferDraw ()

On offer draw message callback.

Returns

the message callback

void onPlannedDisconnect () `[protected]`

On planned disconnect.

MessageCallback onQuestion ()

On question message callback.

Returns

the message callback

MessageCallback onResign ()

On resign message callback.

Returns

the message callback

void onUnplannedDisconnect () `[protected]`

On unplanned disconnect.

MessageCallback onUpdateByMove ()

On update by move message callback.

Returns

the message callback

MessageCallback onUpdateSyncedList ()

On update synced list message callback.

Returns

the message callback

MessageCallback onUsernameAvailability ()

On username availability message callback.

Returns

the message callback

MessageCallback onWaitForMatch ()

On wait for match message callback.

Returns

the message callback

MessageCallback onWaitTurn ()

On wait turn message callback.

Returns

the message callback

MessageCallback onWelcomeMessage ()

On welcome message message callback.

Returns

the message callback

void prepareForDisconnect ()

Prepare for disconnect.

void receivedMessage (Message *message*)

Received message.

Parameters

message	the message

Member Data Documentation

final AppSocket socket`[protected]`

The Socket.

MinimaxMove

Detailed Description

Minimax move.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

MinimaxMove (Move *move*, Evaluation *moveEvaluation*)

Instantiates a new Minimax move.

Parameters

move	the move
moveEvaluation	the move evaluation

MinimaxMove (Evaluation *moveEvaluation*)

Instantiates a new Minimax move.

Parameters

moveEvaluation	the move evaluation

MinimaxMove (MinimaxMove *other*)

Instantiates a new Minimax move.

Parameters

other	the other

Member Function Documentation

int compareTo (MinimaxMove *o*)

Compare to int.

Parameters

o	the o

Returns

the int

boolean equals (Object *o*)

Equals boolean.

Parameters

o	the o

Returns

the boolean

Move getMove ()

Gets move.

Returns

the move

int getMoveDepth ()

Gets move depth.

Returns

the move depth

Evaluation getMoveEvaluation ()

Gets move evaluation.

Returns

the move evaluation

String getShortPrintingStr ()

Gets short printing str.

Returns

the short printing str

int hashCode ()

Hash code int.

Returns

the int

boolean isDeeperAndBetterThan (MinimaxMove *other*)

Is deeper and better than given minimax move.

Parameters

other	the other

Returns

the boolean

void setMove (Move *move*)

Sets move.

Parameters

move	the move

void setMoveEvaluation (Evaluation *moveEvaluation*)

Sets move evaluation.

Parameters

moveEvaluation	the move evaluation

String toString ()

To string string.

Returns

the string

Move

Classes

- enum **MoveFlag**
- enum **ThreefoldStatus**

Detailed Description

Move - represents a "heavy" move. with a lot of info.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Class Documentation

enum ver14::SharedClasses::Game::Moves::Move::ThreefoldStatus

Threefold status - represents a threefold draw status.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Enumerator:

CAN_CLAIM	Can claim threefold status.
CLAIMED	Claimed threefold status.
NONE	None threefold status.

Constructor & Destructor Documentation

Move (Location *movingFrom*, Location *movingTo*, PieceType *capturingPieceType*)

Instantiates a new **Move**.

Parameters

movingFrom	the moving from
movingTo	the moving to
capturingPieceType	the capturing piece type

Move (Location *movingFrom*, Location *movingTo*)

Instantiates a new **Move**.

Parameters

movingFrom	the moving from
movingTo	the moving to

Move (ThreefoldStatus *threefoldStatus*)

Instantiates a new **Move**.

Parameters

threefoldStatus	the threefold status

Move (Move *other*)

Instantiates a new **Move**.

Parameters

other	the other

Member Function Documentation

static Move castling (Location *movingFrom*, Location *movingTo*, CastlingRights.Side *side*) `[static]`

creates a Castling move.

Parameters

movingFrom	the moving from
movingTo	the moving to
side	the castling side

Returns

the move

int compareTo (Move *o*)

Compare to another move.

Parameters

o	the other move

Returns

the comparison result

static Move copyMove (Move *move*) `[static]`

Copy move.

Parameters

move	the move

Returns

the move

boolean equals (Object *o*)

Equals boolean.

Parameters

o	the o

Returns

the boolean

Reimplemented from **BasicMove** (*p.10*).

static Move flipMove (Move *move*)`[static]`

Flip move move.

Parameters

move	the move

Returns

the move

String getAnnotation ()

Gets annotation.

Returns

the annotation

PieceType getCapturingPieceType ()

Gets capturing piece type.

Returns

the capturing piece type

MovesList getCreatorList ()

Gets creator list.

Returns

the creator list

byte getDisabledCastling ()

Gets disabled castling.

Returns

the disabled castling

Location getEnPassantLoc ()

Gets en passant loc.

Returns

the en passant loc

BasicMove getIntermediateMove ()

Gets intermediate move.

Returns

the intermediate move

Evaluation getMoveEvaluation ()

Gets move evaluation.

Returns

the move evaluation

MoveFlag getMoveFlag ()

Gets move flag.

Returns

the move flag

PlayerColor getMovingColor ()

Gets moving color.

Returns

the moving color

int getPrevFullMoveClock ()

Gets prev full move clock.

Returns

the prev full move clock

int getPrevHalfMoveClock ()

Gets prev half move clock.

Returns

the prev half move clock

PieceType getPromotingTo ()

Gets promoting to.

Returns

the promoting to

boolean isCapturing ()

Is capturing boolean.

Returns

the boolean

boolean isCheck ()

Is check boolean.

Returns

the boolean

boolean isReversible ()

Is reversible boolean.

Returns

the boolean

void setCapturing (PieceType *pieceType*)

Sets capturing.

Parameters

pieceType	the piece type

void setCreatorList (MovesList *creatorList*)

Sets creator list.

Parameters

creatorList	the creator list

void setDisabledCastling (byte *disabledCastling*)

Sets disabled castling.

Parameters

disabledCastling	the disabled castling

void setEnPassantLoc (Location *epsnLoc*)

Sets en passant loc.

Parameters

epsnLoc	the epsn loc

void setIntermediateMove (BasicMove *intermediateMove*)

Sets intermediate move.

Parameters

intermediateMove	the intermediate move

void setMoveAnnotation (String *moveAnnotation*)

Sets move annotation.

Parameters

moveAnnotation	the move annotation

void setMoveEvaluation (Evaluation *moveEvaluation*)

Sets move evaluation.

Parameters

moveEvaluation	the move evaluation

void setMoveFlag (MoveFlag *moveFlag*)

Sets move flag.

Parameters

moveFlag	the move flag

void setMovingColor (PlayerColor *movingPlayerColor*)

Sets moving color.

Parameters

movingPlayerColor	the moving player color

void setPrevFullMoveClock (int *prevFullMoveClock*)

Sets prev full move clock.

Parameters

prevFullMoveClock	the prev full move clock

void setPrevHalfMoveClock (int *prevHalfMoveClock*)

Sets prev half move clock.

Parameters

prevHalfMoveClock	the prev half move clock

void setPromotingTo (PieceType *promotingTo*)

Sets promoting to.

Parameters

promotingTo	the promoting to

void setReversible (boolean *reversible*)

Sets reversible.

Parameters

reversible	the reversible

void setThreefoldOption ()

Sets threefold option.

boolean strictEquals (Move *move*)

checks source and destination equals and move flags equals.

Parameters

move	the move

Returns

the boolean

static Move threefoldClaim () `[static]`

Threefold claim move.

Returns

the move

String toString ()

To string string.

Returns

the string

Reimplemented from **BasicMove** (*p.11*).

MoveAnnotation

Detailed Description

Move annotation - utility class that annotates moves.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

static String annotate (Move *move*, Piece *movingPiece*) `[static]`

Annotate move.

Parameters

move	the move
movingPiece	the moving piece

Returns

the annotation

static String annotate (Move *move*, Piece *movingPiece*, String *unique*) `[static]`

Annotate move with a unique string.

Parameters

move	the move
movingPiece	the moving piece
unique	the unique string

Returns

the string

static String basicAnnotate (BasicMove *move*) `[static]`

Basic annotate a move. just the source and destination.

Parameters

move	the move

Returns

the string

Move.MoveFlag Enum Reference

Public Attributes

- **NormalMove**
- **EnPassant**

- **DoublePawnPush**
- **Promotion**
- **ShortCastle**
- **LongCastle**
- final boolean **isCastling**
- final CastlingRights.Side **castlingSide**

Static Public Attributes

- static final **MoveFlag**[] **CASTLING_FLAGS**

Detailed Description

Move flag - which type of move this is.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

MoveFlag ()

Instantiates a new **Move** flag.

MoveFlag (CastlingRights.Side *side*)

Instantiates a new **Move** flag.

Parameters

side	the side

Member Function Documentation

boolean equals (Move *myMove*, Move *otherMove*)

checks if the flags are equals.

Parameters

myMove	the my move
otherMove	the other move

Returns

the boolean

Member Data Documentation

final MoveFlag [] CASTLING_FLAGS `[static]`

The Castling flags.

final CastlingRights.Side castlingSide

if this is a castling flag, which side is it.

DoublePawnPush

Double pawn push move flag.

EnPassant

En passant move flag.

final boolean isCastling

Is this move flag a castling flag.

LongCastle

Long castle move flag.

NormalMove

Normal move move flag.

Promotion

The Promotion.

ShortCastle

Short castle move flag.

MyError

Detailed Description

My error - .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

MyError ()

Instantiates a new My error.

MyError (Throwable *throwable*)

Instantiates a new My error.

Parameters

throwable	the throwable

MyError (String *message*)

Instantiates a new My error.

Parameters

message	the message

MyError (String *message*, Throwable *cause*)

Instantiates a new My error.

Parameters

message	the message
cause	the cause

Member Function Documentation

static String errToString (Throwable *error*)[static]

Err to string string.

Parameters

error	the error

Returns

the string

String getHandledStr ()

Gets handled str.

Returns

the handled str

String toString ()

To string string.

Returns

the string

MyJButton

Detailed Description

MyJButton- represents a button.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

MyJButton (String *text*, VoidCallback *callback*)

Instantiates a new My j button.

Parameters

text	the text
callback	the callback

MyJButton (String *text*)

Instantiates a new My j button.

Parameters

text	the text

MyJButton ()

Instantiates a new My j button.

MyJButton (String *text*, Font *font*, VoidCallback *onClick*, JPanel *addTo*)

Instantiates a new My j button.

Parameters

text	the text
font	the font
onClick	the on click
addTo	the add to

MyJButton (String *text*, Font *font*, VoidCallback *onClick*)

Instantiates a new My j button.

Parameters

text	the text
font	the font
onClick	the on click

MyJButton (String *text*, Font *font*)

Instantiates a new My j button.

Parameters

text	the text
font	the font

Member Function Documentation

int getMinSize ()

Gets min size.

Returns

the min size

void replaceWithCancel (VoidCallback *onCancelled*)

Replace with cancel.

Parameters

onCancelled	the on cancelled

void resetState (boolean *e*)

Reset state.

Parameters

e	the e

void setFont (Font *font*)

Sets font.

Parameters

font	the font

void setOnClick (VoidCallback *onClick*)

Sets on click.

Parameters

onClick	the on click

void setText (String *text*)

Sets text.

Parameters

text	the text

MyJFrame

Classes

- interface **BooleanClosing**
- interface **Closing**
- interface **StringClosing**

Protected Attributes

- **Closing**<?> **onClose**

Detailed Description

The type My j frame.

Constructor & Destructor Documentation

MyJFrame () throws HeadlessException

Instantiates a new My j frame.

Exceptions

HeadlessException	the headless exception

Member Function Documentation

static MyAdapter debugAdapter (Window *addTo*)`[static]`

Debug adapter my adapter.

Parameters

addTo	the add to

Returns

the my adapter

void doXClick ()

Do x click.

MyAdapter getMyAdapter ()

Gets my adapter.

Returns

the my adapter

void setOnExit (Closing<?> *onClose*)

Sets on exit.

Parameters

onClose	the on close

void setOnResize (VoidCallback *onResize*)

Sets on resize.

Parameters

onResize	the on resize

Member Data Documentation

Closing<?> onClose `[protected]`

The On close.

MyThread

Detailed Description

The type My thread.

Constructor & Destructor Documentation

MyThread ()

Instantiates a new My thread.

Member Function Documentation

static void currentThread (Callback< MyThread > *run*) `[static]`

Current thread. will only execute code if inside a **MyThread**

Parameters

run	the run

abstract void handledRun () throws Throwable `[abstract]`, `[protected]`

Handled run.

Exceptions

Throwable	the throwable

Reimplemented in **AppSocket** (*p.2*), and **HandledThread** (*p.54*).

void ignoreErrs ()

Ignore errs.

void reactivateErrs ()

Reactivate errs.

void stopRun ()

Stop run.

ObjCallback< T >

Detailed Description

Parameters

<T>	the type

Piece Enum Reference

Public Attributes

- **W_P**
- **W_R**
- **W_B**
- **W_N**
- **W_Q**
- **W_K**
- **B_P**
- **B_R**
- **B_B**
- **B_N**

- **B_Q**
- **B_K**
- final **PieceType pieceType**
- final **PlayerColor playerColor**

Static Public Attributes

- static final **Piece**[] **ALL_PIECES**

Detailed Description

Piece - represents a combination of a piece type, and a color.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

See also

PieceType

PlayerColor

Member Function Documentation

String getFen ()

Gets fen.

Returns

the fen

static Piece getPiece (PieceType *pieceType*, PlayerColor *playerColor*)`[static]`

Gets piece.

Parameters

pieceType	the piece type
playerColor	the player color

Returns

the piece

static Piece getPieceFromFen (char *c*)`[static]`

Gets piece from fen.

Parameters

c	the c

Returns

the piece from fen

String getPieceIcon ()

Gets piece icon.

Returns

the piece icon

static String getPieceIcon (Piece *piece*)`[static]`

Gets piece icon.

Parameters

piece	the piece

Returns

the piece icon

boolean isOnMyTeam (Piece *otherPiece*)

Is on my team boolean.

Parameters

otherPiece	the other piece

Returns

the boolean

boolean isOnMyTeam (PlayerColor *otherPlayerColor*)

Is on my team boolean.

Parameters

otherPlayerColor	the other player color

Returns

the boolean

Member Data Documentation

final Piece [] ALL_PIECES`[static]`

The constant ALL_PIECES.

B_B

represents a Black Bishop.

B_K

represents a Black King.

B_N

represents a Black Knight.

B_P

represents a Black p piece.

B_Q

represents a Black Queen.

B_R

represents a Black Rook.

final PieceType pieceType

The **Piece** type.

final PlayerColor playerColor

The Player color.

W_B

represents a White Bishop.

W_K

represents a White King.

W_N

represents a White Knight.

W_P

represents a White p piece.

W_Q

represents a White Queen.

W_R

represents a White Rook.

PieceType Enum Reference

Public Attributes

- **PAWN**
- **ROOK**
- **BISHOP**
- **KNIGHT**
- **QUEEN**
- **KING**
- final String **whiteIcon**
- final String **blackIcon**
- final int **value**
- final boolean **isSliding**
- final int **asInt**

Static Public Attributes

- static final int **NUM_OF_PIECE_TYPES**
- static final **PieceType**[] **PIECE_TYPES**
- static final **PieceType**[] **UNIQUE_MOVES_PIECE_TYPES**
- static final **PieceType**[] **MINOR_PIECES**
- static final **PieceType**[] **MAJOR_PIECES**
- static final **PieceType**[] **CAN_PROMOTE_TO**
- static final **PieceType**[] **ATTACKING_PIECE_TYPES**

- static final **Direction**[][] **ATTACKING_DIRECTIONS**

Detailed Description

Piece type - represents the **Piece** Type .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

boolean compareMovementType (PieceType *compareTo*)

Compare movement type boolean.

Parameters

compareTo	the compare to

Returns

the boolean

static boolean compareMovementType (PieceType *piece1Type*, PieceType *piece2Type*)`[static]`

Compare movement type boolean.

Parameters

piece1Type	the piece 1 type
piece2Type	the piece 2 type

Returns

the boolean

Direction[] getAttackingDirections ()

Get attacking directions direction [ ].

Returns

the direction [ ]

static Direction[] getAttackingDirections (PieceType *pieceType*)`[static]`

Get attacking directions direction [ ].

Parameters

pieceType	the piece type

Returns

the direction [ ]

String getPieceIcon (PlayerColor *playerColor*)

Gets piece icon.

Parameters

playerColor	the player color

Returns

the piece icon

String getPieceName ()

Gets piece name.

Returns

the piece name

static PieceType getPieceType (int *pieceType*) `[static]`

Gets piece type.

Parameters

pieceType	the piece type

Returns

the piece type

Direction[] getWalkingDirections ()

Get walking directions direction [ ].

Returns

the direction [ ]

String getWhitePieceFen ()

Gets white piece fen.

Returns

the white piece fen

boolean isAttack (Direction *direction*, int *maxDistance*)

Is attack boolean.

Parameters

direction	the direction
maxDistance	the max distance

Returns

the boolean

static boolean isDiagonalPiece (PieceType *pieceType*) `[static]`

Is diagonal piece boolean.

Parameters

pieceType	the piece type

Returns

the boolean

static boolean isLinePiece (PieceType *pieceType*) `[static]`

Is line piece boolean.

Parameters

pieceType	the piece type

Returns

the boolean

Member Data Documentation

final int asInt

The As int.

final Direction [][] ATTACKING_DIRECTIONS [static]

The Attacking directions. each piece type has its own set of attacking direction.

final PieceType [] ATTACKING_PIECE_TYPES [static]

The constant ATTACKING_PIECE_TYPES.

BISHOP

Bishop piece type.

final String blackIcon

The Black icon.

final PieceType [] CAN_PROMOTE_TO [static]

The types of pieces a pawn Can promote to.

final boolean isSliding

The Is sliding.

KING

The King.

KNIGHT

The Knight.

final PieceType [] MAJOR_PIECES [static]

The Major pieces.

final PieceType [] MINOR_PIECES [static]

The Minor pieces.

final int NUM_OF_PIECE_TYPES [static]

The constant NUM_OF_PIECE_TYPES.

PAWN

Pawn **Piece** Type.

final PieceType [] PIECE_TYPES [static]

The constant PIECE_TYPES.

QUEEN

Queen piece type.

ROOK

Rook piece type.

final PieceType [] UNIQUE_MOVES_PIECE_TYPES `[static]`

The Unique moves piece types.

final int value

The Value.

final String whiteIcon

The White icon.

PlayerColor Enum Reference

Public Attributes

- **WHITE**
- **BLACK**
- **NO_PLAYER**
- final int **asInt**
- final int **indexOf2**
- final int **startingRow**
- final int **diff**

Static Public Attributes

- static final **PlayerColor**[] **PLAYER_COLORS**
- static final int **NUM_OF_PLAYERS**

Detailed Description

Player color - represents a player color.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

PlayerColor (int *startingRow*, int *diff*)

Instantiates a new Player color.

Parameters

startingRow	the starting row
diff	the diff

Member Function Documentation

static PlayerColor getColor (int *clr*) `[static]`

Gets color.

Parameters

clr	the clr

Returns

the color

String getName ()

Gets name.

Returns

the name

abstract PlayerColor getOpponent () `[abstract]`

Gets opponent.

Returns

the opponent

static PlayerColor getPlayerFromFen (String *playerToMove*) `[static]`

Gets player from fen.

Parameters

playerToMove	the player to move

Returns

the player from fen

String toString ()

To string string.

Returns

the string

Member Data Documentation

final int asInt

The As int.

BLACK

Black.

final int diff

The moving up ratio.

final int indexOf2

index of two. calculated at initialization for performance. calculation: **`asInt`** *2

NO_PLAYER

No player.

final int NUM_OF_PLAYERS`[static]`

The constant NUM_OF_PLAYERS.

final PlayerColor [] PLAYER_COLORS`[static]`

The constant PLAYER_COLORS.

final int startingRow

The player's Starting row.

WHITE

White.

PreMadeRequest

Classes

- interface **VariationCreator**

Public Attributes

- final int **authSettings**

Static Public Attributes

- static final **PreMadeRequest TopPlayers**
- static final **PreMadeRequest Games**
- static final **PreMadeRequest DeleteUnfGames**
- static final **PreMadeRequest StatsByTimeOfDay**
- static final **PreMadeRequest ChangeProfilePic**
- static final **PreMadeRequest**[] **statistics**

Detailed Description

Pre made request - .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

RequestBuilder createBuilder ()

Create builder request builder.

Returns

the request builder

PreMadeRequest[] getRequestVariations ()

Get request variations pre made request [ ].

Returns

the pre made request [ ]

Member Data Documentation

final int authSettings

The Auth settings.

final PreMadeRequest ChangeProfilePic `[static]`

The constant ChangeProfilePic.

final PreMadeRequest DeleteUnfGames `[static]`

The constant DeleteUnfGames.

final PreMadeRequest Games `[static]`

The constant Games.

final PreMadeRequest [] statistics `[static]`

The Statistics.

final PreMadeRequest StatsByTimeOfDay `[static]`

The constant StatsByTimeOfDay.

final PreMadeRequest TopPlayers `[static]`

The constant TopPlayers.

Question

Classes

- enum **QuestionType**

Public Attributes

- final String **questionStr**
- final **QuestionType questionType**

Static Public Attributes

- static final **Question Threefold**
- static final **Question Rematch**

Detailed Description

Question.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Class Documentation

enum ver14::SharedClasses::Misc::Question::QuestionType

Question type.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Enumerator:

DRAW_OFFER	Draw offer question type.
NO_TYPE	No type question type.
REMATCH	Rematch question type.
THREEFOLD	Threefold question type.

Constructor & Destructor Documentation

Question (String *questionStr*, Answer... *possibleAnswers*)

Instantiates a new **Question**.

Parameters

questionStr	the question str
possibleAnswers	the possible answers

Question (String *questionStr*, QuestionType *questionType*, Answer... *possibleAnswers*)

Instantiates a new **Question**.

Parameters

questionStr	the question str
questionType	the question type
possibleAnswers	the possible answers

Member Function Documentation

record Answer (String *answerStr*)

Answer.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

static Question drawOffer (String *offeringPlayer*)`[static]`

Draw offer question.

Parameters

offeringPlayer	the offering player

Returns

the question

boolean equals (Object *o*)

Equals boolean.

Parameters

o	the o

Returns

the boolean

Answer getDefaultAnswer ()

Gets default answer.

Returns

the default answer

Answer[] getPossibleAnswers ()

Get possible answers answer [ ].

Returns

the answer [ ]

String getQuestionStr ()

Gets question str.

Returns

the question str

void setDefaultAnswer (Answer *defaultAnswer*)

Sets default answer.

Parameters

defaultAnswer	the default answer

String toString ()

To string string.

Returns

the string

Member Data Documentation

final String questionStr

The **Question** str.

final QuestionType questionType

The **Question** type.

final Question Rematch `[static]`

The constant Rematch.

final Question Threefold `[static]`

The constant Threefold.

RegEx

Public Attributes

- final String[] **dontMatch**

Static Public Attributes

- static final **RegEx Fen**
- static final **RegEx Username**
- static final **RegEx Password**
- static final **RegEx Icon**
- static final **RegEx StrUtilSkip**
- static final **RegEx Numbers**
- static final **RegEx URL**
- static final **RegEx DontSaveGame**
- static final **RegEx IPPAddress**
- static final **RegEx Any**

Detailed Description

Reg ex.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

RegEx (@Language("RegExp") String *regEx*, String *details*, String... *dontMatch*)

Instantiates a new Reg ex.

Parameters

regEx	the reg ex
details	the details
dontMatch	the dont match

RegEx (@Language("RegExp") String *regEx*, String *details*, boolean *useDontMatch*, String... *dontMatch*)

Instantiates a new Reg ex.

Parameters

regEx	the reg ex
details	the details
useDontMatch	the use dont match
dontMatch	the dont match

Member Function Documentation

RegEx canBeEmpty (boolean bool , String *emptyDetails*)

Can be empty reg ex.

Parameters

bool	the bool
emptyDetails	the empty details

Returns

the reg ex

boolean check (String *str*)

Check boolean.

Parameters

str	the str

Returns

the boolean

RegEx get (boolean *useDontMatch*)

Get reg ex.

Parameters

useDontMatch	the use dont match

Returns

the reg ex

String getDetails ()

Gets details.

Returns

the details

String getRegex ()

Gets regex.

Returns

the regex

static boolean isSavedDate (String *str*)`[static]`

Is saved date boolean.

Parameters

str	the str

Returns

the boolean

void setUseDontMatch (boolean *useDontMatch*)

Sets use dont match.

Parameters

useDontMatch	the use dont match

Member Data Documentation

final RegEx Any `[static]`

The constant Any.

final String [] dontMatch

The Dont match.

final RegEx DontSaveGame `[static]`

The constant DontSaveGame.

final RegEx Fen `[static]`

The constant Fen.

final RegEx Icon `[static]`

The constant Icon.

final RegEx IPPAddress `[static]`

The constant IPPAddress.

final RegEx Numbers `[static]`

The constant Numbers.

final RegEx Password `[static]`

The constant Password.

final RegEx StrUtilSkip `[static]`

The constant StrUtilSkip.

final RegEx URL `[static]`

The constant URL.

final RegEx Username `[static]`

The constant Username.

RequestBuilder

Public Attributes

- final **Arg**[] **args**

Static Public Attributes

- static final String **TIE_STR**

Protected Attributes

- final **SQLStatement statement**
- final String **name**
- String **postDescription**
- String **preDescription**
- **RequestBuilder subBuilder**

Detailed Description

Request builder - creates builders capable of generating complete sql statements. after building with the required arguments .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

RequestBuilder (DBRequest *request*, PreMadeRequest.Variation *variation*)

Instantiates a new Request builder.

Parameters

request	the request
variation	the variation

RequestBuilder (SQLStatement *statement*, String *name*, Arg... *args*)

Instantiates a new Request builder.

Parameters

statement	the statement
name	the name
args	the args

RequestBuilder (SQLStatement *statement*, String *name*, String *desc*, Arg... *args*)

Instantiates a new Request builder.

Parameters

statement	the statement
name	the name
desc	the desc
args	the args

RequestBuilder (SQLStatement *statement*, String *name*, String *postDescription*, String *preDescription*, Arg... *args*)

Instantiates a new Request builder.

Parameters

statement	the statement
name	the name
postDescription	the post description
preDescription	the pre description
args	the args

Member Function Documentation

void addShouldSync (SyncedListType *listType*)`[protected]`

Add should sync.

Parameters

listType	the list type

DBRequest build (Object... *argsVals*)

Build db request.

Parameters

argsVals	the args vals

Returns

the db request

static RequestBuilder changePassword ()`[static]`

Change password request builder.

Returns

the request builder

static RequestBuilder changeProfilePic ()`[static]`

Change profile pic request builder.

Returns

the request builder

DBResponse createResponse (ResultSet *rs*, DBRequest *request*)

Create response db response.

Parameters

rs	the rs
request	the request

Returns

the db response

static RequestBuilder createVariation (ObjCallback< RequestBuilder > *og*, PreMadeRequest.VariationCreator *variationCreator*)`[static]`

Create variation request builder.

Parameters

og	the og
variationCreator	the variation creator

Returns

the request builder

static RequestBuilder deleteAllUnFinishedGames () `[static]`

Delete all un finished games request builder.

Returns

the request builder

static RequestBuilder games () `[static]`

Games request builder.

Returns

the request builder

Arg[] getArgs ()

Get args arg [ ].

Returns

the arg [ ]

String getArgVal (int *index*)

Gets arg val.

Parameters

index	the index

Returns

the arg val

String getName ()

Gets name.

Returns

the name

String getPostDescription ()

Gets post description.

Returns

the post description

String getPreDescription ()

Gets pre description.

Returns

the pre description

ArrayList< SyncedListType > getShouldSync ()

Gets should sync.

Returns

the should sync

void setSubBuilder (RequestBuilder *subBuilder*)

Sets sub builder.

Parameters

subBuilder	the sub builder

static RequestBuilder statsByTimeOfDay () `[static]`

Stats by time of day request builder.

Returns

the request builder

static RequestBuilder top () `[static]`

Top request builder.

Returns

the request builder

Member Data Documentation

final Arg [] args

The request Arguments.

final String name `[protected]`

The Name.

String postDescription `[protected]`

The Post description.

String preDescription `[protected]`

The Pre description.

final SQLStatement statement `[protected]`

The Statement.

RequestBuilder subBuilder `[protected]`

The Sub builder.

final String TIE_STR `[static]`

The constant TIE_STR.

Selection

Classes

- interface **Join**
- interface **Order**

Additional Inherited Members

Detailed Description

Selection - a selection sql statement.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Class Documentation

interface ver14::SharedClasses::DBActions::Statements::Selection::Join

Join - selection join.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Class Members:

String	LEFT	The constant LEFT.

interface ver14::SharedClasses::DBActions::Statements::Selection::Order

Order - selection order by.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Class Members:

String	ASC	The constant ASC.
String	DESC	The constant DESC.

Constructor & Destructor Documentation

Selection (Object *selectFrom*, Object[] *select*)

Instantiates a new **Selection**.

Parameters

selectFrom	the select from
select	the select

Selection (Object *selectFrom*, Condition *condition*, Object[] *select*)

Instantiates a new **Selection**.

Parameters

selectFrom	the select from
condition	the condition
select	the select

Member Function Documentation

String createStatement () [protected]

Create statement string.

Returns

the string

Reimplemented from **SQLStatement** (*p.120*).

void join (@Join String *joinType*, Table *joinWith*, Condition *condition*, Col... *groupBy*)

Join.

Parameters

joinType	the join type
joinWith	the join with
condition	the condition
groupBy	the group by

Selection nestMe (Col... *outerSelect*)

Nest me selection.

Parameters

outerSelect	the outer select

Returns

the selection

void orderBy (Col *col*, @Order String *order*)

Order by.

Parameters

col	the col
order	the order

void top (Object *top*)

Top.

Parameters

top	the top

CastlingRights.Side Enum Reference

Public Attributes

- **KING**
- **QUEEN**
- final int **rookStartingCol**
- final int **castledRookCol**
- final int **castledKingCol**

- final int **kingTravelDistance**
- final String **castlingNotation**
- final int **asInt**
- final int **mult**

Static Public Attributes

- static final **Side**[] **SIDES**

Detailed Description

Side - Castling side.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Side (int *castledKingCol*, int *rookStartingCol*, int *castledRookCol*)

Instantiates a new **Side**.

Parameters

castledKingCol	the castled king col
rookStartingCol	the rook starting col
castledRookCol	the castled rook col

Member Function Documentation

Location kingFinalLoc (Location *currentKingLoc*)

King final loc location.

Parameters

currentKingLoc	the current king loc

Returns

the location

String toString ()

To string string.

Returns

the string @hidden

Member Data Documentation

final int asInt

The As int.

final int castledKingCol

The Castled king col.

final int castledRookCol

The Castled rook col.

final String castlingNotation

The Castling notation.

KING

King side.

final int kingTravelDistance

The King travel distance.

final int mult

The King Movement **Direction** Mult

QUEEN

Queen side.

final int rookStartingCol

The Rook starting col.

final Side [] SIDES `[static]`

The constant SIDES.

GameStatus.SpecificStatus Enum Reference

Public Attributes

- **Checkmate**
- **TimedOut**
- **TimedOutVsInsufficientMaterial**
- **Resignation**
- **GameGoesOn**
- **ThreeFoldRepetition**
- **Stalemate**
- **InsufficientMaterial**
- **FiftyMoveRule**
- **TieByAgreement**
- **PlayerDisconnectedVsAi**
- **PlayerDisconnectedVsReal**
- **ServerStoppedGame**
- final **GameStatusType gameStatusType**

Detailed Description

Specific status - specific game status .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

SpecificStatus ()

Instantiates a new Specific status.

SpecificStatus (GameStatusType *gameStatusType*)

Instantiates a new Specific status.

Parameters

gameStatusType	the game status type

Member Function Documentation

String toString ()

To string string.

Returns

the string

Member Data Documentation

Checkmate

Checkmate.

FiftyMoveRule

Fifty move rule.

GameGoesOn

Game goes on .

final GameStatusType gameStatusType

The Game status type.

InsufficientMaterial

Insufficient material.

PlayerDisconnectedVsAi

The Player disconnected vs ai.

PlayerDisconnectedVsReal

The Player disconnected vs real.

Resignation

Resignation.

ServerStoppedGame

Server stopped game.

Stalemate

Stalemate.

ThreeFoldRepetition

Three fold repetition.

TieByAgreement

The Tie by agreement.

TimedOut

Timed out.

TimedOutVsInsufficientMaterial

Timed out vs insufficient material.

SQLStatement

Public Attributes

- final DBRequest.Type **type**

Detailed Description

Sql statement - represents an sql statement.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

SQLStatement (DBRequest.Type *type*)

Instantiates a new Sql statement.

Parameters

type	the type

Member Function Documentation

abstract String createStatement () `[abstract]`, `[protected]`

Create statement string.

Returns

the string

Reimplemented in **CustomStatement** (*p.31*), **Delete** (*p.35*), **Selection** (*p.116*), and **Update** (*p.137*).

String getStatement ()

Gets statement.

Returns

the statement

void replace (String *replacing*, String *replaceWith*)

Replace.

Parameters

replacing	the replacing
replaceWith	the replace with

String toString ()

To string string.

Returns

the string

Member Data Documentation

final DBRequest.Type type

The request Type.

Square

Static Public Attributes

- static final **Piece EMPTY_PIECE**

Detailed Description

Square represents a square on the logic board.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Square (Location *loc*)

Instantiates a new **Square**.

Parameters

loc	the loc

Square (Piece *piece*, Location *loc*)

Instantiates a new **Square**.

Parameters

piece	the piece
loc	the loc

Member Function Documentation

String getFen ()

Gets fen.

Returns

the fen

Location getLoc ()

Gets loc.

Returns

the loc

Piece getPiece ()

Gets piece.

Returns

the piece

String getPieceIcon ()

Gets piece icon.

Returns

the piece icon

boolean isEmpty ()

Is empty boolean.

Returns

the boolean

void setEmpty ()

Sets empty.

void setPiece (Piece *piece*)

Sets piece.

Parameters

piece	the piece

String toString ()

To string string.

Returns

the string

Member Data Documentation

final Piece EMPTY_PIECE [static]

The constant EMPTY_PIECE.

StatusResponse

Additional Inherited Members

Detailed Description

Status response - .

Author

Bezalel Avrahami (bezalel3250@gmail.com)

Constructor & Destructor Documentation

StatusResponse (Status *status*, DBRequest *request*, int *updatedRows*)

Instantiates a new Status response.

Parameters

status	the status
request	the request
updatedRows	the updated rows

StatusResponse (Status *status*, String *details*, DBRequest *request*, int *updatedRows*)

Instantiates a new Status response.

Parameters

status	the status
details	the details
request	the request
updatedRows	the updated rows

Member Function Documentation

DBResponse clean ()

Clean db response.

Returns

the db response

Reimplemented from **DBResponse** (*p.33*).

String getDetails ()

Gets details.

Returns

the details

boolean isAnyData ()

Is any data boolean.

Returns

the boolean

Reimplemented from **DBResponse** (*p.34*).

MyJFrame.StringClosing

Additional Inherited Members

Detailed Description

The interface String closing.

Member Function Documentation

default boolean checkClosingVal (String *val*)

Check closing val boolean.

Parameters

val	the val

Returns

the boolean

default String show ()

Show string.

Returns

the string

StrUtils

Classes

- interface **IterationThingy**

Detailed Description

Str utils.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

static String awful (String *og*) `[static]`

Awful string.

Parameters

og	the og

Returns

the string

static String clean (String *str*) `[static]`

Clean string.

Parameters

str	the str

Returns

the string

static int countMatches (String *str*, @Language("RegExp") String *match*) `[static]`

Count matches int.

Parameters

str	the str
match	the match

Returns

the int

static String createTimeGreeting () `[static]`

Create time greeting string.

Returns

the string

static String createTimeStr (long *millis*) `[static]`

Create time str string.

Parameters

millis	the millis

Returns

the string

static String dateTimePrefix (String *str*) `[static]`

Date time prefix string.

Parameters

str	the str

Returns

the string

static String dontCapFull (String *str*)[static]

Dont cap full string.

Parameters

str	the str

Returns

the string

static String dontCapWord (String *str*)[static]

Dont cap word string.

Parameters

str	the str

Returns

the string

static String fitInside (String *str*, JComponent *comp*)[static]

Fit inside string.

Parameters

str	the str
comp	the comp

Returns

the string

static String fixHtml (String *str*)[static]

Fix html string.

Parameters

str	the str

Returns

the string

static String format (String *str*)[static]

Format string.

Parameters

str	the str

Returns

the string

static String[] format (String[] *arr*)[static]

Format string [ ].

Parameters

arr	the arr

Returns

the string [ ]

static String[][] format (String *mat*[][]) `[static]`

Format string [ ] [ ].

Parameters

mat	the mat

Returns

the string [ ] [ ]

static String formatDate (Date *date*) `[static]`

Format date string.

Parameters

date	the date

Returns

the string

static String formatDate (Date *date*, String *format*) `[static]`

Format date string.

Parameters

date	the date
format	the format

Returns

the string

static String formatDate (String *longStr*) `[static]`

Format date string.

Parameters

longStr	the long str

Returns

the string

static String formatDateSQL (Date *date*) `[static]`

Format date sql string.

Parameters

date	the date

Returns

the string

static int getPort (Socket *socket*) `[static]`

Gets port.

Parameters

socket	the socket

Returns

the port

static int getPort (SocketAddress *socketAddress*)`[static]`

Gets port.

Parameters

socketAddress	the socket address

Returns

the port

static String getUrl (Socket *socket*)`[static]`

Gets url.

Parameters

socket	the socket

Returns

the url

static String getUrl (SocketAddress *socketAddress*)`[static]`

Gets url.

Parameters

socketAddress	the socket address

Returns

the url

static String htmlNewLines (String *str*)`[static]`

Html new lines string.

Parameters

str	the str

Returns

the string

static boolean isEmpty (String *str*)`[static]`

Is empty boolean.

Parameters

str	the str

Returns

the boolean

static void main (String[] *args*)`[static]`

The entry point of application.

Parameters

args	the input arguments

static String parseURLS (String *str*)`[static]`

Parse urls string.

Parameters

str	the str

Returns

the string

static String repeat (IterationThingy< String > *iterationThingy*, int *numOfIterations*) `[static]`

Repeat string.

Parameters

iterationThingy	the iteration thingy
numOfIterations	the num of iterations

Returns

the string

static String splitArr (Object[] *arr*) `[static]`

Split arr string.

Parameters

arr	the arr

Returns

the string

static String splitArr (String *divide*, Object[] *arr*) `[static]`

Split arr string.

Parameters

divide	the divide
arr	the arr

Returns

the string

static String splitArr (String *divide*, Object[] *arr*, boolean *format*) `[static]`

Split arr string.

Parameters

divide	the divide
arr	the arr
format	the format

Returns

the string

static String strINN (Object... *objs*) `[static]`

Str inn string.

Parameters

objs	the objs

Returns

the string

static String uppercase (String *str*) `[static]`

Uppercase string.

Parameters

str	the str

Returns

the string

SwitchCase

Detailed Description

Switch case - represents a case that is meant to be used inside a switch case col. if the **`condition`** is true, the **`ifTrue`** col will display in the switch case col

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

SwitchCase (Condition *condition*, Col *ifTrue*)

Instantiates a new Switch case.

Parameters

condition	the condition
ifTrue	the if true

Member Function Documentation

Condition condition ()

Condition condition.

Returns

the condition

static SwitchCase defaultCase (Col *ifTrue*)`[static]`

Default case switch case.

Parameters

ifTrue	the if true

Returns

the switch case

static SwitchCase equals (Col *col*, String *value*, Col *ifTrue*)`[static]`

Equals switch case.

Parameters

col	the col
value	the value
ifTrue	the if true

Returns

the switch case

Col ifTrue ()

If true col.

Returns

the col

String toString ()

To string string.

Returns

the string

Table Enum Reference

Public Attributes

- **Games**
- **UnfinishedGames**
- **Users**
- final **Col**[] **cols**

Detailed Description

Table - represents the tables in the db.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Table (Col... *cols*)

Instantiates a new **Table**.

Parameters

cols	the cols

Member Function Documentation

static String escapeValues (Object[] *values*, boolean *quotes*, boolean *parentheses*) `[static]`

Escape values string.

Parameters

values	the values
quotes	quotes

parentheses	parentheses

Returns

the escaped string

String tableAndValues ()

Table and values string.

Returns

the string

Member Data Documentation

final Col [] cols

The Cols in this table

Games

Games table.

UnfinishedGames

Unfinished games table.

Users

Users table.

TableDBResponse

Protected Attributes

- String[] **columns**
- String[][] **rows**

Detailed Description

Table db response - .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

TableDBResponse (String[] *columns*, String *rows*[][], DBRequest *request*)

Instantiates a new Table db response.

Parameters

columns	the columns
rows	the rows

request	the request

TableDBResponse (String[] *columns*, String *rows*[][], Status *status*, DBRequest *request*)

Instantiates a new Table db response.

Parameters

columns	the columns
rows	the rows
status	the status
request	the request

TableDBResponse () `[protected]`

Instantiates a new Table db response.

Member Function Documentation

TableDBResponse clean ()

Clean table db response.

Returns

the table db response

Reimplemented from **DBResponse** (*p.33*).

String getCell (int *row*, Col *col*)

Gets cell.

Parameters

row	the row
col	the col

Returns

the cell

String getCell (int *row*, String *col*)

Gets cell.

Parameters

row	the row
col	the col

Returns

the cell

int getColumnIndex (String *column*) `[protected]`

Gets column index.

Parameters

column	the column

Returns

the column index

String[] getColumns ()

Get columns string [ ].

Returns

the string [ ]

String[] getFirstRow ()

Get first row string [ ].

Returns

the string [ ]

String[][] getRows ()

Get rows string [ ] [ ].

Returns

the string [ ] [ ]

boolean isAnyData ()

Is any data boolean.

Returns

the boolean

Reimplemented from **DBResponse** (*p.34*).

int numOfRows ()

Num of rows int.

Returns

the int

String rowToString (String[] *row*)

Row to string string.

Parameters

row	the row

Returns

the string

String toString ()

To string string.

Returns

the string

Reimplemented from **DBResponse** (*p.34*).

Member Data Documentation

String [] columns`[protected]`

The Columns.

String [][] rows [protected]

The Rows.

ThreadsManager

Detailed Description

The type Threads manager.

Member Function Documentation

static MyThread createThread (ThrowingRunnable *runnable*, boolean *start*) [static]

Create thread my thread.

Parameters

runnable	the runnable
start	the start

Returns

the my thread

static void handleErrors (ThrowingRunnable *runnable*) [static]

Handle errors.

Parameters

runnable	the runnable

ThrowingCallback< T >

Detailed Description

Throwing callback - a callback that might throw an exception .

Parameters

<T>	the callback type

Author

Bezalel Avrahami (bezalel3250@gmail.com)

Member Function Documentation

void callback (T *obj*) throws Exception

Callback.

Parameters

obj	the obj

Exceptions

Exception	the exception

UnfinishedGame

Public Attributes

- final **PlayerColor playerColorToMove**
- final String **playerToMove**

Additional Inherited Members

Detailed Description

The type Unfinished game.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

UnfinishedGame (String *gameId*, String *creatorUsername*, GameSettings *gameSettings*, String *opponentUsername*, PlayerColor *playerColorToMove*, String *playerToMove*, Stack< Move > *moveStack*)

Instantiates a new Unfinished game.

Parameters

gameId	the game id
creatorUsername	the creator username
gameSettings	the game settings
opponentUsername	the opponent username
playerColorToMove	the player color to move
playerToMove	the player to move
moveStack	the move stack

Member Function Documentation

boolean isCreatorToMove ()

Is creator to move boolean.

Returns

the boolean

Member Data Documentation

final PlayerColor playerColorToMove

The Player color to move.

final String playerToMove

The Player to move.

Update

Additional Inherited Members

Detailed Description

Update - a sql update statement.

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

Update (Table *updating*, Condition *condition*, NewValue... *newValues*)

Instantiates a new **Update**.

Parameters

updating	the updating
condition	the condition
newValues	the new values

Member Function Documentation

String createStatement () `[protected]`

Create statement string.

Returns

the string

Reimplemented from **SQLStatement** (*p.120*).

ValueBtn< T >

Detailed Description

Value btn - a value holding button.

Parameters

<T>	the value's type

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Constructor & Destructor Documentation

ValueBtn (String *text*, Font *font*, T *value*, Callback< T > *onClick*)

Instantiates a new Value btn.

Parameters

text	the text
font	the font
value	the value
onClick	the on click

PreMadeRequest.VariationCreator

Detailed Description

Variation creator - .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

Variation create (RequestBuilder *actualBuilder*)

Create variation.

Parameters

actualBuilder	the actual builder

Returns

the variation

VoidCallback

Detailed Description

Void callback - .

Author

Bezalel Avrahami (`bezalel3250@gmail.com`)

Member Function Documentation

void callback ()

Callback.

Index

INDEX