

---

# Class ConcurrentTCPServerThread

```
java.lang.Object
|
+-- java.lang.Thread
|
+-- ConcurrentTCPServerThread
```

## All Implemented Interfaces:

java.lang.Runnable

---

< [Constructors](#) > < [Methods](#) >

---

```
public class ConcurrentTCPServerThread
extends java.lang.Thread
```

## Constructors

### ConcurrentTCPServerThread

```
public ConcurrentTCPServerThread(java.net.Socket _sock)
```

## Methods

### run

```
public void run()
```

#### Overrides:

run in class java.lang.Thread

---

# Class ConcurrentUDPServerThread

```
java.lang.Object
|
+-- java.lang.Thread
|
+-- ConcurrentUDPServerThread
```

## All Implemented Interfaces:

java.lang.Runnable

---

< [Constructors](#) > < [Methods](#) >

---

```
public class ConcurrentUDPServerThread
extends java.lang.Thread
```

## Constructors

### ConcurrentUDPServerThread

```
public ConcurrentUDPServerThread(java.net.DatagramSocket _dgSocket)
```

## Methods

### run

```
public void run()
```

#### Overrides:

run in class java.lang.Thread

---

## Class Database

```
java.lang.Object
|
+--Database
```

---

< [Methods](#) >

---

```
public class Database
extends java.lang.Object
```

Database class for Gossip P2P Server

#### Author:

sroig2013@my.fit.edu

## Methods

## getInstance

```
public static synchronized Database getInstance()
```

Access the database singleton object.

**Returns:**

Return instance of the Database.

---

## getPeersList

```
public PeersListMessage getPeersList()
```

Queries list of all known peers in the db.

**Returns:**

List of all known peers.

---

## handleGossipMessage

```
public boolean handleGossipMessage(GossipMessage _gossipMessage)
```

Checks db for Gossip Message. Inserts if message doesn't exist. Updates record if it does exist.

**Parameters:**

\_gossipMessage - Message to search for in db.

**Returns:**

---

## handlePeerMessage

```
public boolean handlePeerMessage(PeerMessage _peerMessage)
```

Checks db for peer message. Inserts if message doesn't exist. Updates record if it does exist.

**Parameters:**

\_peerMessage - The message to update or insert into db.

**Returns:**

Returns true if query is successful. False if not.

---

## initializeDatabase

```
public void initializeDatabase(java.lang.String _pathToDB)
    throws java.sql.SQLException
```

Setup database using the path passed in from args.

**Parameters:**

\_pathToDB - Path to the database.

**Throws:**

java.sql.SQLException -

---

## Class ErrorMessage

```
java.lang.Object
|
+-- Message
    |
    +-- ErrorMessage
```

---

< [Constructors](#) >

---

```
public class ErrorMessage
    extends Message
```

Represents an invalid message.

**Author:**

sroig2013@my.fit.edu

## Constructors

### ErrorMessage

```
public ErrorMessage()
```

---

## Class GossipMessage

```
java.lang.Object
|
+-- Message
    |
    +-- GossipMessage
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class GossipMessage  
extends Message
```

Represents a GOSSIP message received by the server

**Author:**

sroig2013@my.fit.edu

## Constructors

### GossipMessage

```
public GossipMessage()
```

Default constructor

---

### GossipMessage

```
public GossipMessage(java.lang.String _sha256EncodedMessage,  
                     java.lang.String _messageDate,  
                     java.lang.String _message)
```

Constructs new GossipMessage.

**Parameters:**

\_sha256EncodedMessage - SHA256 encoding of message.

\_messageDate - Date of message.

\_message - Contents of message.

---

### GossipMessage

```
public GossipMessage(java.lang.String _sha256EncodedMessage,  
                     java.lang.String _messageDate,  
                     java.lang.String _message,  
                     int _id)
```

Constructs new GossipMessage.

**Parameters:**

\_sha256EncodedMessage - SHA256 encoding of message.

\_messageDate - Date of message.

\_message - Contents of message.

---

## GossipMessage

```
public GossipMessage(java.lang.String _sha256EncodedMessage,  
                     java.util.Date _messageDate,  
                     java.lang.String _message)
```

Constructs new GossipMessage.

**Parameters:**

\_sha256EncodedMessage - SHA256 encoding of message.

\_messageDate - Date of message.

\_message - Contents of message.

---

## GossipMessage

```
public GossipMessage(java.lang.String _sha256EncodedMessage,  
                     java.util.Date _messageDate,  
                     java.lang.String _message,  
                     int _id)
```

Constructs new GossipMessage.

**Parameters:**

\_sha256EncodedMessage - SHA256 encoding of message.

\_messageDate - Date of message.

\_message - Contents of message.

\_id - Id assigned from db.

## Methods

### getId

```
public int getId()
```

**Returns:**

Gossip Message id.

---

### getMessage

```
public java.lang.String getMessage()
```

**Returns:**

Gossip Message message.

---

## getMessageDate

```
public java.util.Date getMessageDate()
```

**Returns:**

Gossip Message date.

---

## getSha256EncodedMessage

```
public java.lang.String getSha256EncodedMessage()
```

**Returns:**

Gossip Message SHA 256 encrypted message.

---

## toString

```
public java.lang.String toString()
```

String representation of GossipMessage.

**Overrides:**

toString in class java.lang.Object

---

# Class GossipP2PServer

```
java.lang.Object
|
+--GossipP2PServer
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class GossipP2PServer
extends java.lang.Object
```

Gossip P2P Server

**Author:**

sroig2013@my.fit.edu

## Constructors

## GossipP2PServer

```
public GossipP2PServer()
```

## Methods

### main

```
public static void main(java.lang.String[] args)
```

## Class IterativeTCPServerThread

```
java.lang.Object
|
+-- java.lang.Thread
|
+-- IterativeTCPServerThread
```

### All Implemented Interfaces:

java.lang.Runnable

---

< [Constructors](#) > < [Methods](#) >

---

```
public class IterativeTCPServerThread
extends java.lang.Thread
```

Thread for an iterative TCP Gossip Server.

### Author:

sroig2013@my.fit.edu

## Constructors

### IterativeTCPServerThread

```
public IterativeTCPServerThread(int _port)
```

Construct a new iterative TCP Gossip Server.

### Parameters:

\_port - Port that server will listen on.

## Methods



## run

```
public void run()
```

Run the thread.

**Overrides:**

run in class java.lang.Thread

---

# Class IterativeUDPServerThread

```
java.lang.Object
|
+-- java.lang.Thread
|   |
|   +-- IterativeUDPServerThread
```

**All Implemented Interfaces:**

java.lang.Runnable

---

< [Constructors](#) > < [Methods](#) >

---

```
public class IterativeUDPServerThread
extends java.lang.Thread
```

Thread for an iterative UDP Gossip Server.

**Author:**

sroig2013@my.fit.edu

## Constructors

### IterativeUDPServerThread

```
public IterativeUDPServerThread(int _port)
```

Construct a new iterative UDP Gossip Server.

**Parameters:**

\_port - Port that server will listen on.

## Methods

## run

```
public void run()
```

Run the thread.

**Overrides:**

run in class java.lang.Thread

---

# Class Message

```
java.lang.Object
|
+--Message
```

**Direct Known Subclasses:**

[ErrorMessage](#), [GossipMessage](#), [PeerMessage](#), [PeersListMessage](#)

---

< [Constructors](#) > < [Methods](#) >

---

```
public class Message
extends java.lang.Object
```

Super class that represents messages received by the server

**Author:**

Santiago Roig

## Constructors

### Message

```
public Message()
```

## Methods

### identifyMessage

```
public static Message identifyMessage(java.lang.String strMessage)
```

Identifies message type and returns new instance based on message type.

**Parameters:**

strMessage - String message to be identified.

**Returns:**

Returns a message that has been identified by strMessage.

---

# Class MessageHandler

```
java.lang.Object
|
+--MessageHandler
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class MessageHandler
extends java.lang.Object
```

Class to handle actions for specific message types.

**Author:**

sroig2013@my.fit.edu

## Constructors

### MessageHandler

```
public MessageHandler()
```

## Methods

### HandleGossipMessage

```
public static void HandleGossipMessage(Message _gossipMessage)
```

Handles gossip messages. Store message in database and broadcast it to all known peers.

**Parameters:**

\_gossipMessage - Gossip message to handle.

---

### HandlePeerMessage

```
public static void HandlePeerMessage(Message _peerMessage)
```

Handles peer message. Sends message to database to be processed.

**Parameters:**

\_peerMessage - Peer message to handle.

---

# Class PeerMessage

```

java.lang.Object
|
+--Message
    |
    +--PeerMessage
  
```

< [Constructors](#) > < [Methods](#) >

public class **PeerMessage**  
 extends [Message](#)

Message that will add a new peer to the list of known peers

**Author:**

sroig2013@my.fit.edu

## Constructors

### PeerMessage

```

public  PeerMessage(java.lang.String _peerName,
                    int _portNumber,
                    java.lang.String _ipAddress)
  
```

Constructs a new peer message.

**Parameters:**

\_peerName -  
 \_portNumber -  
 \_ipAddress -

### PeerMessage

```

public  PeerMessage(java.lang.String[] _peerMessage)
  
```

Constructs new Peer Message.

**Parameters:**

\_peerMessage - Array of strings to be turned into peer message.

## Methods

## getId

```
public int getId()
```

**Returns:**

Peer Message id.

---

## getIpAddress

```
public java.lang.String getIpAddress()
```

**Returns:**

Peer Message IP Address.

---

## getPeerName

```
public java.lang.String getPeerName()
```

**Returns:**

Peer Message peer name.

---

## getPortNumber

```
public int getPortNumber()
```

**Returns:**

Peer Message port number.

---

# Class PeersListMessage

```
java.lang.Object
|
+-- Message
|
+-- PeersListMessage
```

---

< [Constructors](#) > < [Methods](#) >

---

```
public class PeersListMessage
extends Message
```

Represents list of known peers.

**Author:**

sroig2013@my.fit.edu

## Constructors

### PeersListMessage

```
public PeersListMessage()
```

---

### PeersListMessage

```
public PeersListMessage(java.util.ArrayList _peers)
```

Constructs new list of peers.

**Parameters:**

\_peers - List of known peers.

## Methods

### getPeers

```
public java.util.ArrayList getPeers()
```

**Returns:**

List of known peers.

---

### toString

```
public java.lang.String toString()
```

String representation of Peers List.

**Overrides:**

toString in class java.lang.Object