TESTING:
Stabilita connessione a server php socket (ReactPHP) con telnet ma la connessione non era predisposta alla connessione tramite la classe WebSocket di JavaScrip, passaggioa all'utilizzo di Ratchet, una libreria che incapsula il sistema di socket di ReactPHP con un sistema più userfriendly e con la gestione del protocollo utilizzato da WebSocket.
Creato un server di test che rimanda lo stesso messaggio ricevuto dal client allo stesso

```
class MyServer implements MessageComponentInterface {
    public function onOpen(ConnectionInterface $conn) {
        echo "Connesso";
    }
    public function onMessage(ConnectionInterface $from, $msg) {
        echo $msg;
        $from->send($msg);
    }
    public function onClose(ConnectionInterface $conn) {
        echo "Disconnesso";
    }
    public function onError(ConnectionInterface $conn, \Exception $e) {
        $conn->close();
    }
}
$server = IoServer::factory(
    new HttpServer(
        new WsServer(
            new MyServer()
        )
    ),
    8000
);
$server->run();
```

Dopo un test tra macchine diverse sulla stessa rete, un errore è stato lanciato perché la connessione non era sicura, ma inizializzata all'interno di una connessione che era già sicura (Server Apache https), allora è stata creata la reverse proxy dal server Apche al server delle socket mantenendo così la connessione sotto lo stesso certificato.

Classe Room testa aggiungendo ogni client alla stessa stanza e provando ad inviare i messaggi tramite questa a tutti i client collegati.

TODO:
Per rendere il sistema ancora più facile da implementare si potrebbe ristrutturare il sistema per incapsulare
Sviluppatore: Marco Anghinetti