

Forum:[Hack Forums](#)

Thread: [Imap conecting error](#)

smtpz - July 30, 2020, 1:32 pm

Since 1 week i got this error when is try to connect to imap

```
at emitReadable_ (_stream_readable.js:555:12)
  at process._tickCallback (internal/process/next_tick.js:63:19) type: 'no', textCode: 'LIMIT', source:
'protocol' }
{ Error: Timed out while authenticating with server
  at Timeout._onTimeout (/check_smtp_bot/node_modules/imap/lib/Connection.js:139:17)
  at ontimeout (timers.js:436:11)
  at tryOnTimeout (timers.js:300:5)
  at listOnTimeout (timers.js:263:5)
  at Timer.processTimers (timers.js:223:10) source: 'timeout-auth' }
{ Error: Timed out while authenticating with server
  at Timeout._onTimeout (/check_smtp_bot/node_modules/imap/lib/Connection.js:139:17)
```

Exalted - July 31, 2020, 3:21 am

Code Timed out while authenticating with server
Could you share some code?

smtpz - July 31, 2020, 6:41 am

checkMail.js

Code

.....

```
class checkMail{

  sendMail(data){
    var hash = db.encrypt(data.user);
    .....
    let transporter = nodemailer.createTransport({
      host: data.ip,
      port: 25,
      secure: false, // true for 465, false for other ports
      ignoreTLS : true,
      auth: {
        user: data.user, // generated ethereal user
        pass: data.pass // generated ethereal password
      }
    });
    var mailOptions = {
      from: data.user,
      to: config.imap.user,
      subject: hash,
      text: 'That was easy!'
```

```

};
transporter.sendMail(mailOptions,function(error , info){
  if(error){
    logger.info(error);
    var updateObj = {
      'response_by' : 'uid0',
      'status' : 2,
      'seen' : 1 ,
      'response': "Product refunded!"
    }
    .....
    .....
    db.Query(sql,function(userInfo){
      var sql = "update users set balance=balance+"+
userInfo[0].price+" ,money_refounded=money_refounded+"+ userInfo[0].price +" where username="
+ userInfo[0].reported_by +""";
      db.customInsert(sql, function(){
        logger.info('Money Refunded to user : '+ userInfo[0].reported_by)
      });
    })
  }
  else{
    logger.info('mail sended with '+ data.user +' with smtp port 25');
    data.time = date().format('HH:mm');
    data.hash = hash;
    feed.setFeed(data.id,data)
    logger.info(info.response);
  }
});
}

else if((data.type == 'u587') || (data.type == '587smtp')){
  logger.info('sending mail with user '+ data.user +' with smtp port 587');
  let transporter = nodemailer.createTransport({
    host: data.ip,
    port: 587,
    ignoreTLS : true,
    secure: false, // true for 465, false for other ports
    auth: {
      user: data.user, // generated ethereal user
      pass: data.pass // generated ethereal password
    }
  });
  var mailOptions = {
    from: data.user,
    to: config.imap.user,
    subject: hash,
    text: 'That was easy!'
  };
  transporter.sendMail(mailOptions,function(error , info){
    if(error){

```

```

logger.info(error);
var updateObj = {
  'response_by' : 'bot',
  'status' : 2,
  'seen' : 1 ,
  'response': "Product refunded!"
}
db.update(data.id,updateObj,'reports',function(done){
  .....
  db.Query(sql,function(userInfo){
    var sql = "update users set balance=balance+"+
userInfo[0].price+" ,money_refounded=money_refounded+"+ userInfo[0].price +" where username="
+ userInfo[0].reported_by +"";
    db.customInsert(sql, function(){
      logger.info('Money Refunded to user : '+ userInfo[0].reported_by)
    });
  })
})
}
else{
  logger.info('mail sented with '+ data.user +' with smtp port 25');
  data.time =date().format('HH:mm');
  data.hash = hash;
  feed.setFeed(data.id,data)
  logger.info(info.response);
}
});
}

```

```

else if((data.type == 'ussl') || (data.type == 'sslsmtplib')){
  logger.info('sending mail with user '+ data.user +' with smtp port 465');
  let transporter = nodemailer.createTransport({
    host: data.ip,
    port: 465,
    tls:{
      rejectUnauthorized: false
    },
    secure: true, // true for 465, false for other ports
    auth: {
      user: data.user, // generated ethereal user
      pass: data.pass // generated ethereal password
    }
  });
  var mailOptions = {
    from: data.user,
    to: config.imap.user,
    subject: hash,
    text: 'That was easy!'
  };
  transporter.sendMail(mailOptions,function(error , info){
    if(error){
      logger.info(error);
      var updateObj = {

```

```

        'response_by' : 'uid0',
        'status' : 2,
        'seen' : 1 ,
        'response': "Product refunded!"
    }
    db.update(data.id,updateObj,'reports',function(done){
        .....
    .....
        db.Query(sql,function(userInfo){
            var sql = "update users set balance=balance++"+
userInfo[0].price+" ,money_refounded=money_refounded++"+ userInfo[0].price +" where username="
+ userInfo[0].reported_by +"";
            db.customInsert(sql, function(){
                logger.info('Money Refunded to user : '+ userInfo[0].reported_by)
            });
        })
    })
}
else{
    logger.info('mail sended with '+ data.user +' with smtp port 465');
    data.time = date().format('HH:mm');
    data.hash = hash;
    feed.setFeed(data.id,data)
    logger.info(info.response);
}
});
}
else{
    logger.info('the user :'+ data.user +' is not valid' );
}
}

readMail(obj,done){
    var self = this;
    var imap = new Imap({
        user: config.imap.user,
        password: config.imap.password,
        host: config.imap.host,
        port: config.imap.port,
        tls: true,
    });

    function openInbox(mailbox,cb) {
        imap.openBox(mailbox, true, cb);
    }
    imap.once('ready', function() {
        openInbox('INBOX',function(err, box) {
            var mails = [];
            if (err) throw err;
            imap.search(['ALL',['SUBJECT', obj.hash ]], function(err, results) {
                if(results && results.length > 0){
                    var updateObj = {

```

```

        'response_by' : 'uid0',
        'status' : 3,
        'seen' : 1 ,
        'response': "Refound rejected!"
    }
    db.update(obj.id,updateObj,'reports',function(res){
        logger.info('this report is rejected submitted by :'+ obj.user)
        done(res)
    });
}
else{
    logger.info('NO Mail Match In Inbox');
    logger.info('Start read messages from spam folder');
    imap.closeBox(function(err){
        openInbox('Bulk Mail',function(err, box) {
            imap.search(['ALL', ["SUBJECT", obj.hash ]], function(err, spamResult) {
                if(spamResult && spamResult.length > 0){
                    var updateObj = {
                        'response_by' : 'uid0',
                        'status' : 3,
                        'seen' : 1 ,
                        'response': "Refound rejected!"
                    }
                    db.update(obj.id,updateObj,'reports',function(res){
                        logger.info('this report is rejected submitted by :'+ obj.user)
                        //done(res)
                    });
                }
            }
        }
        else{
            var updateObj = {
                'response_by' : 'uid0',
                'status' : 2,
                'seen' : 1 ,
                'response': "Product refunded!"
            }
            db.update(obj.id,updateObj,'reports',function(res){
                logger.info('this report is correct the Product is refunded reported by : '+ obj.user)
                var sql = "select r.reseller_response as reseller_response,p.id as id_product,p.add_by
as add_by,r.id as id,p.type as type,p.ip as ip,p.user as user,p.pass as pass,r.status as
status,r.date_sent as date_sent,r.proof as proof,r.date_response as date_response,r.response as
response,p.price as price,r.response_by as response_by,r.user as reported_by from products
p,reports r where r.id_product=p.id and r.id= '"+obj.id;
                db.Query(sql,function(userInfo){
                    var sql = "update users set balance=balance+"+
userInfo[0].price+"",money_refounded=money_refounded+" "+ userInfo[0].price +" where username="
+ userInfo[0].reported_by +"";
                    db.customInsert(sql, function(){
                        logger.info('Money Refunded to user : '+ userInfo[0].reported_by)
                    });
                });
            });
        }
    });
}
}
});

```

```

    });
  })
}
});
});
});
imap.once('error', function(err) {
  console.log(err);
});
imap.once('end', function() {
  console.log('Connection ended');
});
imap.connect();
}
}
module.exports = new checkMail;

```

config.js

```
Code var config = {};
```

```

config.server = {};
config.imap = {};
config.db = {};
config.cron = {};

```

```
config.server.port = 3001;
```

```
config.cron.time = '*/2 * * * *';
```

```

config.db.server = 'localhost';
config.db.user = 'user';
config.db.password = 'database';
config.db.schema = 'pass';
config.db.port = '3306';

```

```

config.imap.user = 'someemail@yahoo.com';
config.imap.password = 'fgthftghjhuere';
config.imap.host = 'imap.mail.yahoo.com';
config.imap.port = 993 ;
config.imap.tsl = true;
// config.imap.user = 'someemail@gmail.com';
// config.imap.password = 'nchhknqujfnhtvea';
// config.imap.host = 'imap.gmail.com';
// config.imap.port = 993 ;
// config.imap.tls = true;

```

```
module.exports = config;
```