

C

Node Document

Kill Port :- `sudo kill -9 $(sudo lsof -t -i:9001) => Suppose 9001 is the port Number`

To get timezone offset

```
const date = new Date();  
const offset = date.getTimezoneOffset();  
console.log(offset);
```

After that save the another timestamp in db and show accordingly

```
let currenttime = Math.round(new Date().getTime() / 1000);  
let exatctime = currenttime + (offset);
```

```
sequelize-auto -h localhost create_user= await model.create({  
  email: requestdata.email,  
  name : requestdata.name,  
  password:confirm_password,  
  device_type:requestdata.device_type,  
  device_token:requestdata.device_token,  
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users  
  
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

CMD ubuntu

```
- sudo apt-get install node
- sudo apt-get install npm
- npm init
- npm install express --save
- npx express-generator
- express --view=ejs
- npm install --save express-fileupload --save
- npm install --save path --save npm install sequelize --save
- npm install sequelize --save
-
- npm install crypto
-
- npm install sequelize --save
- npm install sequelize --save
- npm install express-session
- npx sequelize-cli init
- Or
- -
```

```
sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
```

```
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

```

- npm install -g sequelize-auto -save
- npm install -g mysql --save

- npm install mysql2 --save

- npm install mysql2 --save

- Socket io max listener: require('events').EventEmitter.prototype._maxListeners = 1000;

```

```

on: {coll: sequelize.where(sequelize.col('`booking`.user_id'), '=', sequelize.col('review.provider_id')),},

```

```

//if location and lat lng not working on modal then just add below code in style

```

```

div.pac-container { z-index: 1050 !important; }

```

```

stripe_connect_doc:- https://stripe.com/docs/connect/standard-accounts#token-request

```

```

Kill port :- sudo killall node ,

```

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form

```

Live server connect : `sudo ssh -i /pem-file-path servername@hostname`

Increase Watcher In Terminal

```
Increase Watcher : echo fs.inotify.max_user_watches=582222 | sudo tee -a /etc/sysctl.conf && sudo sysctl -p
```

```
Set global sql_mode='';
```

Database query create model

1:

Eg:

1. `sequelize-auto -h localhost -d test -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t ,order, -C`

This is work :-

New eg: `sequelize-auto -h localhost -d dbname-u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t tablerna`

Eg: `sequelize-auto -o "/var/www/html/chatobite/db/models" -d chatobite -h localhost -u dbuser -x cqlsys123 -e mysql -t userDetails`

```
sequelize-auto -h localhost create_user= await model.create({  
  email: requestdata.email,  
  name : requestdata.name,  
  password:confirm_password,  
  device_type:requestdata.device_type,  
  device_token:requestdata.device_token,
```

`t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users`

`sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form`

```
sequelize-auto -o "/var/www/html/mastoo/models" -d msToo_db -h localhost -e mysql -t ms_login
```

2: sudo apt install nodejs npm

Queries:-----

1: Find one query:-----

```
const users = await model.tablename.findOne({
  where : {
    email : requestdata.email
  },
  Order:[
    ['id','desc']
  ]
});
```

2: insert Query

```
const      auth_id:auth_create,
```

```
sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
```

```
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

```
    image:imageName
  });
```

3: Update Query-----

```
let save = await model.tablename.update({
  name:req.omniserv body.categoryname
},
{
  where: {
    id: req.body.id
  }
})
```

4: delete Query -----

```
const dlt = await cate.destroy({
  where: {
    id: req.body.id
  }
});
```

5 or condition

```
const Sequelize = require('sequelize');
const Op = Sequelize.Op;
```

```
sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
```

```
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

```

var user_data = await constants.findAll({
  where : {

    [Op.or]: [{sender_id: msg.sender_id ,reciever_id: msg.receiver_id}, {reciever_id: msg.receiver_id,sender_id: msg.sender_id}]
  }

});

```

Password Set:-----

```

const confirm_password = crypto.createHash('md5').update(requestdata.password).digest('hex');

```

Multiple Array set json form-----

```

GivenIllneses = Illness.split(",").map(
);

```

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,

t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form

```

Image Upload-----

L

**** Route ****

```
App.route('/delete_user').delete(admin.deleterow);
```

*** Function in Controller ****

```
deleterow: async function (req, res){  
  var model=req.body.table;  
  var id=req.body.id;  
  try {  
    await db[model].destroy({  
      where: {  
        id: id  
      }  
    });  
  } catch (e) {
```

```
sequelize-auto -h localhost create_user= await model.create({  
  email: requestdata.email,  
  name : requestdata.name,  
  password:confirm_password,  
  device_type:requestdata.device_type,  
  device_token:requestdata.device_token,
```

```
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```



```

        }

        res.json(1);
    },

```

**** Keep This code in footer ****

```
<script>
```

```

$(document).ready(function () {
    $('.delete').click(function () {
        let _this = $(this);

        let id = $(this).attr("rel");
        let table = $(this).attr("table");

        swal({
            title: "Are you sure?",
            text: "Once deleted, you will not be able to recover this data!",
            icon: "warning",

```

```

sequelize-auto -h localhost create_user= await model.create({
    email: requestdata.email,
    name : requestdata.name,
    password:confirm_password,
    device_type:requestdata.device_type,
    device_token:requestdata.device_token,

```

```
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

```

    buttons: true,
    dangerMode: true,
  })
  .then((data) => {
    if (data) {
      $.ajax({
        url: "/delete_user",
        type: "DELETE",
        data: { id: id, table: table },
        cache: false,
        success: function (response) {

          _this.parents('tr').remove();
          swal("Proof! Your data has been deleted!", {
            icon: "success",
          });
          location.reload();
          /*$(this).load(function() {
            console.log('hii');
          });
        }
      );
    }
  });

```

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,

```

```

t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form

```

```

        setTimeout(function () { location.reload(1); }, 1000);*/
    }, error(error) {
        console.error(error);
    }

    });
}
});
});
});
</script>

```

**** Delete Button ****

```

<ul class="dropdown-menu dropdown-menu-right">
<li class="delete" table="users" rel="<%= response[i].id%>"><a href="#"><i class="icon-trash delete"></i>
        Delete</a></li>
</ul>

```

**** From here pass id and table from where to delete the entry ****

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form

```

**** Sequelizeize Joins ****

```
User.belongsTo(Admin,{foreign_key: 'adminId'});
```

Then Include the attributes where you fire the queries

```
const list= await User.findAll({
  include : [{
    model : Admin ,
    attributes : ['email']
  }],
});
```

Here i am fetching admin email with all users details and adminId is foreign_key in users table

**** Map the response into a easy response form ****

Here data comes in list

```
var final_data= list.map(user => {
  return user.toJSON();
});
```

```
sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
```

```
t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form
```

```
});
```

**** Joining In Sequelizeize ****

First Make Relation among all table

```
User.belongsTo(Admin,{foreign_key: 'adminId'});
```

**** Then Include ****

```
include : [{  
  model : Admin ,  
  attributes : ['email']  
}],
```

Include Conditon

```
include: [{ model: User, as: 'Owner', where: { username: username} }]
```

```
sequelize-auto -h localhost create_user= await model.create({  
  email: requestdata.email,  
  name : requestdata.name,  
  password:confirm_password,  
  device_type:requestdata.device_type,  
  device_token:requestdata.device_token,
```

```
t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form
```

Date to string

```
Code: var dateString = response.dataValues.createdAt;
dateString = new Date(dateString).toUTCString();
dateString = dateString.split(' ').slice(0, 4).join(' ');
```

Example:-

Fri Jul 05 2019 11:41:08 GMT+0530 (IST) to Fri, 05 Jul 2019

BelongsTo :- define top of controller (same as another method)

```
modelName.belongsTo(AnotherModelName, {
  foreignKey: 'modelNameId'
});
```

Query:

```
var users = await user.findOne({
  where : {
    id : req.query.id
  },
  include: [{
    model: brand,}],
```

```
sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
```

```
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

```

order: [
  attributes: ['name','id'],
  required: true

  ['id','desc'],
]
});

```

And and not eql condition

```

var users = await user.findAll({
  where :
    {
      [Op.and]: [
        {
          topSearch: {q
            [Op.ne]: 0
          }
        },

```

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,

```

```

t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form

```

```

    {
      type: {
        [Op.eq]: 0
      }
    },
    {
      status: {
        [Op.eq]: 1
      }
    }
  ]
},

```

how to truncate table

```

const clear = Location.destroy({
  where: {},
  truncate: true
})

```

Foreach :

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form

```



```
_data.forEach(function(item){  
  });
```

Sum columns data

```
var total_admin_Ocount = await invoice.findAll({  
  attributes: [[sequelize.fn('sum', sequelize.col('admin_fees')), 'total']],  
});
```

Custom Query Sequelize:-

Find and create(if duplicate data insert then)

```
const obj = {  
  endoresedUserId: data.endoresedUserId,  
  endoresedBy: data.endoresedBy,  
  type: t  
};
```

```
await UserEndrosment.findOrCreate({  
  where: obj, // we search for this
```

```
sequelize-auto -h localhost create_user= await model.create({  
  email: requestdata.email,  
  name : requestdata.name,  
  password:confirm_password,  
  device_type:requestdata.device_type,  
  device_token:requestdata.device_token,
```

```
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

```
defaults: obj // if not found then insert this
});
```

Create timestamp

```
const timestamp = parseInt(new Date().getTime() / 1000, 10);
```

Validation:-

Server file

```
var expressValidator = require('express-validator');
Version 5.3.1 only
app.use(expressValidator());
```

uValidation file

```
req.check('email').isEmail().withMessage('Email field is required');
req.check('password').notEmpty().withMessage('Password field is required');
```

Include under condition

```
sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

```

model: Users,
attributes: ['roleId'],
on: {
coll: sequelize.where(sequelize.col("postComments.userId"), "=", sequelize.col("postComments->user.id")),
},

```

Group by query in sequelize

```

var providers = await provider.findAll({

  include: [{
    model: ratings,
    attributes: ['id', 'userId', 'providerId', 'orderId', [sequelize.fn('AVG', sequelize.col('rating')),
'avgrating'], 'description'],
    required: false,
  }],
  where: {
    isApprove: 1
  },

```

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,

t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form

```

```

group: [sequelize.col(proconsole.log(get_block_status_data,"get_block_status_data");vider.id)],

order: [
  ['id', 'desc'],
]
});
if (providers) {
  providers = providers.map(value => {
    return value.toJSON();
  });
}

```

Important link

<https://stackoverflow.com/questions/9321225/using-dynamic-search-parameters-with-sequelize-js>

Distance Query:-

<https://gis.stackexchange.com/questions/31628/find-points-within-a-distance-using-mysql>

EG: `var get_driver_online = await driver.sequelize.query('SELECT id,first_name,last_name,latitude,longitude,email,address,avail, (3959 * acos (cos (radians('+data.latitude+')) * cos(`

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,

```

`t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users`

`sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form`

```

radians( latitude ) ) * cos( radians( longitude ) - radians('+data.longitude+') ) + sin ( radians('+data.latitude+') ) * sin(
radians( latitude ) ) ) AS distance FROM drivers where avail = 1 HAVING distance < 100 ORDER BY distance',{
  type: sequelize.QueryTypes.SELECT
});

```

Distance Query Sequelize

```

var get_home_data = await providerCategory.findAll({
  include: [
    {
      model: user,
      required: false,
      attributes: ['id', 'orderId', 'price', 'created', [sequelize.literal('\`provider\`.\`name\`'), 'providerName'],
[sequelize.literal('\`provider\`.\`id\`'), 'providerId'], [sequelize.literal('\`provider\`.\`image\`'), 'providerImage'],
[sequelize.literal('\`provider\`.\`description\`'), 'providerDescription'], [sequelize.fn('Round', sequelize.fn('AVG',
sequelize.col('\`rating\`.\`ratings\`'))), 'ratingAvg'], [sequelize.fn('Round', sequelize.literal("3959 *
acos(cos(radians('+check_auth.dataValues.latitude+')) * cos(radians(provider.latitude)) *
cos(radians('+check_auth.bdataValues.longitude+') - radians(provider.longitude)) +
sin(radians('+check_auth.dataValues.latitude+') * sin(radians(provider.latitude)))"), 2), 'distance']] ,

    // where: {

```

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password: confirm_password,
  device_type: requestdata.device_type,
  device_token: requestdata.device_token,
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form

```

```

        //   categoryId:get_data.category_id
        // },
    },
],

having: { 'order.distance': { [sequelize.Op.lte]: '100' } }
});

```

Average rating with round

```

get_rating_profile= await rating.findOne({
  attributes: ['id', [sequelize.fn('Round', sequelize.col('rating'))], 'ratingAvg']],
  where: {
    user2Id: contrator_data.contractor_id,
    type: 0
  },
  group: ['user2Id']
});

```

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form

```

Cutter audio video code

```
const ffmpeg = require('ffmpeg');
var shell = require('shelljs');
var filePath = process.cwd() + '/public/uploads/' + video_name;
    // console.log(filePath); return;
var filename = Math.floor(Date.now() / 1000) + video_name;
var destPath = process.cwd() + '/public/uploads/' + filename;
var gh = "ffmpeg -i " + filePath + " -ss 00:00:01 -t 00:00:30 -async 1 " + destPath;
    // console.log(gh, "makeCom====="); return;
await shell.exec(gh);
```

sequelize-auto -h localhost create_user= await model.create({

email: requestdata.email,
name : requestdata.name,
password:confirm_password,
device_type:requestdata.device_type,
device_token:requestdata.device_token,

t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form

To Remove file from folder

```
let filePath = path.join('public/images/users/')+fileName;  
fs.unlinkSync(filePath);
```

Apple push Implementation:

```
var apn = require('apn');
```

```
var options = {  
  token: {  
  
    key: __dirname + "/AuthKey_YD72V8ZVPS.p8",  
    keyId: "YD72V8ZVPS",  
    teamId: "6TL3WCKA9D"  
  },  
  production: false  
};
```

```
let bundle_id = 'com.live.FilmGig';  
var apnProvider = new apn.Provider(options);
```

```
sequelize-auto -h localhost create_user= await model.create({  
  email: requestdata.email,  
  name : requestdata.name,  
  password:confirm_password,  
  device_type:requestdata.device_type,  
  device_token:requestdata.device_token,  
t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form
```



```

var send_message = {
    to: _get_data.device_token,

    notification: {
        title: 'Filmgig',
        body: 'Checkout The New Gig',
        sound: "default",
        badge: get_read_data[i].count_data
    },

    data: {

        body: 'Checkout The New Gig',

    }
};

```

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,

```

```

t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form

```

```

var note = new apn.Notification();

// note.expiry = Math.floor(Date.now() / 1000) + 3600; // Expires 1 hour from now.
note.badge = get_read_data[i].count_data;
note.sound = "ping.aiff";
note.alert = "Checkout The New Gig";
note.payload = send_message;
note.topic = bundel_id;
apnProvider.send(note, _get_data.device_token).then((result) => {
    res(send_message);
    console.log(result, "send_message")
});

```

```

sequelize-auto -h localhost create_user= await model.create({

```

```

  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,

```

```

t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form

```

Between Condition:-

```
where:{
  rate:{
    [Op.between]:[requestdata.price_min,requestdata.price_max]
  },
  accountType:1
}
```

Get Month start and end date

```
var month = requestdata.month * 1000;
var startOfMonth = moment(month).startOf('month').format('x') / 1000 ;
var endOfMonth = moment(month).endOf('month').format('x') / 1000;
```

Pagination Logic

```
sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form
```

Routes

```
app.route('/allmatches').get(Matchctrl.allmatches);  
app.route('/allmatches/:id').get(Matchctrl.allmatches);
```

In Controller Function

```
var limit =1;  
var page =req.params.id || 1 ;  
let offset= (limit*page)-limit;  
let details= model.findAndCountAll({  
  Where{  
    condition...  
  },  
  Offset:offset,  
  limit :limit  
});  
  
let allRecord= details.count;  
let pages= allRecord/limit;  
  
And send pages in render pages,  
Totalpagesneed:pages
```

```
sequelize-auto -h localhost create_user= await model.create({  
  email: requestdata.email,  
  name : requestdata.name,  
  password:confirm_password,  
  device_type:requestdata.device_type,  
  device_token:requestdata.device_token,  
})  
  
t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form
```

In Ejs Page

```
<div class="col-md-12">
  <ul class="pagination">
    <li><a href="/allmatches/"id="previous_button">Previous</a></li>
    <%for(i=1;i<=totalpagesneed;i++){%>
      <li><a href="/allmatches/<%=i%>"><%= i %></a></li>
    <%}%>
    <li><a href="/allmatches/" id="next_button">Next</a></li>
  </ul>
</div>
```

In Ejs page IN Script tag

```
<script>
  const currentUrl = window.location.href;
  const currentUrlArray = currentUrl.split('/');
  const currentPage = parseInt(currentUrlArray[currentUrlArray.length - 1]);

  sessionStorage.setItem('currentPage', currentPage);
```

```
sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
})
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

```

let totalPages = <%= totalpagesneed %>;

if (currentPage > 1) {
    $('#previous_button').attr('href', `/allmatches/${currentPage-1}`);
} else {
    $('#previous_button').attr('href', `/allmatches/1`);
}

if (currentPage < totalPages) {
    $('#next_button').attr('href', `/allmatches/${currentPage+1}`);
} else {
    $('#next_button').attr('href', `/allmatches/${totalPages}`);
}
</script>

```

Get Chat with Case Condition :-

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form

```

```

var get_message = await chat.findAll({
  attributes: ['id', 'userId', 'user2Id', 'message', 'constantId', 'type', 'created_at', [ sequelize.literal('CASE WHEN
chat.type = 1 THEN (select name from user where id=chat.user2id) WHEN chat.type = 0 THEN (select name from provider
where id=chat.user2id ) end '), "username" ], [ sequelize.literal('CASE WHEN chat.type = 1 THEN (select image from user
where id=chat.user2id) WHEN chat.type = 0 THEN (select image from provider where id=chat.user2id ) end '), "image" ]],
  where: {
    constantId: constant_check.id
  }
});

```

Convert files like(pdf to html or pdf to docs);

```

const convert = require('ebook-convert');
var xtend = require('xtend')
If calibre-ebook not found your server
Cmd:  sudo -v && wget -nv -O- https://wnload.calibre-ebook.com/linux-installer.sh | sudo sh /dev/stdin

```

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form

```

```
// convert file-----

var name = Date.now() + '.pdf';
var options = {
  input: path.join(__dirname,      '../public/assets/file/'+data.files.file.name),
  output: path.join(__dirname,'../public/assets/file/'+name ),
};
convert(options, function (err) {

  if (err) console.log(err);
});
let filePath = path.join('public/assets/file/')+data.files.file.name;
fs.unlinkSync(filePath);
```

Literal query

```
attributes:['id','name','description','price','offerPrice',
  [sequelize.literal('(select count(*) from backpack where product_id=products.id)'),
'backpack'],[sequelize.literal('(select count(*) from product_comment where product_id=products.id)'),
'comment'],[sequelize.literal('(select count(*) from product_like where product_id=products.id)'), 'like_count']
```

```
sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users
```

```
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```



```
],
```

Add days

```
var today = moment();  
var tomorrow = moment(today).add(1, 'days').format('x') / 1000;
```

Create automatic timestamp without function just copy this code in your model

```
hooks : {  
  beforeCreate : (record, options) => {  
    record.dataValues.createdAt = Math.round(new Date().getTime() / 1000);  
    record.dataValues.updatedAt = Math.round(new Date().getTime() / 1000);  
  },  
  beforeUpdate : (record, options) => {  
    record.dataValues.updatedAt = Math.round(new Date().getTime() / 1000);  
  }  
}
```

GET URL Dynamic

```
sequelize-auto -h localhost create_user= await model.create({  
  email: requestdata.email,  
  name : requestdata.name,  
  password:confirm_password,  
  device_type:requestdata.device_type,  
  device_token:requestdata.device_token,  
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users  
  
sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

```

fullurl:function(req){
  return url.format({
    // protocol: req.protocol,
    host: req.get('host'),
    // pathname: req.originalUrl
  });
},

```

Http to https server

```

const options = {
  key: fs.readFileSync("/etc/letsencrypt/live/dragonsumo.com/privkey.pem"),
  ca: fs.readFileSync("/etc/letsencrypt/live/dragonsumo.com/chain.pem"),
  cert: fs.readFileSync("/etc/letsencrypt/live/dragonsumo.com/cert.pem")
};

```

```

sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

```

```

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form

```

```
const server = require('https').createServer(options, app) ;
```

Issue with mysql phpmyadmin

XAMPP - MySQL shutdown unexpectedly

Show this error when mysql start on win platform

```
12:19:12 PM [mysql] Attempting to start MySQL app...
12:19:12 PM [mysql] Status change detected: running
12:19:13 PM [mysql] Status change detected: stopped
12:19:13 PM [mysql] Error: MySQL shutdown unexpectedly.
12:19:13 PM [mysql] This may be due to a blocked port, missing dependencies,
```

<https://stackoverflow.com/questions/18022809/xampp-mysql-shutdown-unexpectedly>

```
sequelize-auto -h localhost create_user= await model.create({
  email: requestdata.email,
  name : requestdata.name,
  password:confirm_password,
  device_type:requestdata.device_type,
  device_token:requestdata.device_token,
t -d viiu -u dbuser -x cqlsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqlsys123 --dialect mysql -c config/config.json -o models -t form
```

git config --global alias.ac 'commit -a -m'

Hit this command now we don't need to do git add . then git commit -m "message"

Now hit this

git ac "new commit "

git push origin master

Use this command to save creds on git lab and after this you don't have to type password again

git config --global credential.helper store

sequelize-auto -h localhost create_user= await model.create({

email: requestdata.email,

name : requestdata.name,

password:confirm_password,

device_type:requestdata.device_type,

device_token:requestdata.device_token,

t -d viiu -u dbuser -x cqsys123 -p 3306 --dialect mysql -c ./config/config.json -o ./models/ -t users

sequelize-auto -h localhost -d form_ajax -u dbuser -x cqsys123 --dialect mysql -c config/config.json -o models -t form