ChartJS and Vue - tracking growth of plant (or height of child or weight of kitten etc.)

https://vue-chartjs.org/

vue create plant-growth-chart

make the following files:

create new file package.json with

npm init

enter package name
enter description
entry point server.js
author you

Now you have a new file called package.json

npm install express
npm install body-parser
npm install sequelize
npm install pg   # need if you plan to deploy this to heroku
npm install --save-dev sqlite3

package.json looks like this

```
{
  "name": "plant-height-records",
  "version": "1.0.0",
  "description": "Plant height with chart",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "author": "clara",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.19.0",
    "express": "^4.17.1",
    "pg": "^8.5.1",
    "sequelize": "^6.3.5"
  },
  "devDependencies": {
    "sqlite3": "^5.0.0"
  }
}
```

server.js [copy in from other project] https://github.com/claraj/student-sign-in-vue-express-api/blob/master/server.js rename line 8 to use your project name
routes directory
   - api.js [empty for now]
models directory
  - index.js [copy from other project and change student to plantRecord
  - record.js [empty for now]

config**.json** [copy from other project] https://github.com/claraj/student-sign-in-vue-express-api/blob/master/
config.json rename "storage" from "student.sqlite" to "plant.sqlite" or appropriate for your concept

models/record.js

```
module.exports = (sequelize, DataTypes) => {

    let PlantRecord = sequelize.define('PlantRecord', {
        height: {
            type: DataTypes.NUMBER,
            allowNull: false
        }, date: {
            type: DataTypes.DATE,
            allowNull: false
        }
    })

    PlantRecord.sync({force: true}).then( () => {
        console.log('synced plant record table')
    })
    return PlantRecord
}
```

routes/api.js

```
let express = require('express')
let db = require('../models')  // imports index.js from models
directory, which
let PlantRecord = db.PlantRecord

let router = express.Router()

// get all records
router.get('/plant_records', function(req, res, next){
    PlantRecord.findAll({order: ['date']}).then( plantRecords =>
{
        return res.json(plantRecords)
    }).catch( err => next(err) )
})

// add new record
router.post('/plant_records', function(req, res, next) {
    PlantRecord.create(req.body).then( () => {
        res.status(201).send('ok')
    }).catch( next(err) )  // todo send more specific error
message
})

module.exports = router
```

Run your server

127.0.0.1:3000/api/plant_records

Expect empty array

curl, add and query

```
curl http://127.0.0.1:3000/api/plant_records
```

```
curl -X POST --data '{"height": 1, "date":
"2012-04-23T18:25:43.511Z"}' http://127.0.0.1:3000/api/
plant_records -H "Content-Type: Application/JSON"
```

All OK? Change sync: false to prevent recreating DB tables

Vue code!

cd plant-growth-chart

```
npm install chart.js         <-- yes dot  .js
npm install vue-chartjs      <-- no dot
npm install axios
```

Open App.vue and HelloWord.vue
Rename HelloWorld and references to it to PlantChart - check App.vue

https://github.com/claraj/student-sign-in-vue-express-api/blob/master/student-client/vue.config.js

Create new file vue.config.js in your plant-growth-chart

Restart vue dev server after saving

new directory

src/services/PlantService.js

```
import axios from 'axios'

export default {

    getAllPlantRecords() {
        return axios.get('/api/plant_records').then( response =>
{
            return response.data
        })
    },
    addPlantRecord(plantRecord) {
        return axios.post('/api/plant_records',
plantRecord).then( response => {
            return response.data
        })
    }

}
```

src/main.js

```
import Vue from 'vue'
import App from './App.vue'
import PlantService from '@/services/PlantService'

Vue.prototype.$plant_record_api = PlantService

Vue.config.productionTip = false

new Vue({
  render: h => h(App),
}).$mount('#app')
```

App.vue - a form, some data, method to add new record

```
<template>
  <div id="app">

    <label>Height</label>
    <input type="number" v-model="newHeight">
    <br>
    <label>Date</label>
    <input type="date" v-model="newDate">
    <br>
    <button v-on:click="addRecord">Add record</button>

    <plant-chart/>  <!--todo -->
  </div>
</template>

<script>
import PlantChart from './components/PlantChart.vue'

export default {
  name: 'App',
  components: {
    PlantChart
  },
  data() {
    return {
      newHeight: 0,
      newDate: '',
    }
  },
  methods: {
    addRecord() {
      // todo validation
      let record = {height: this.newHeight, date: new
Date(this.newDate)}
      console.log(record)
      this.
$plant_record_api.addPlantRecord(record).then( response => {
        // todo update chart
      })
    }
}
}
</script>

<style>
```

```
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

Try adding data - check at 127.0.0.1:3000/api/plant_records

Is data sorted by date? Earliest first

Let's get data

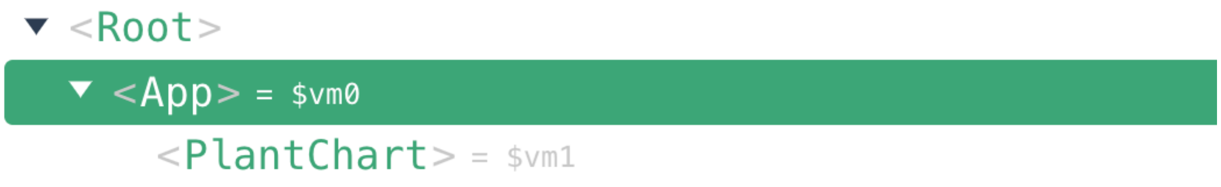
```
<template>
  <div id="app">

    <label>Height</label>
    <input type="number" v-model="newHeight">
    <br>
    <label>Date</label>
    <input type="date" v-model="newDate">
    <br>
    <button v-on:click="addRecord">Add record</button>
```

```
    <plant-chart/>  <!--todo -->
  </div>
</template>
```

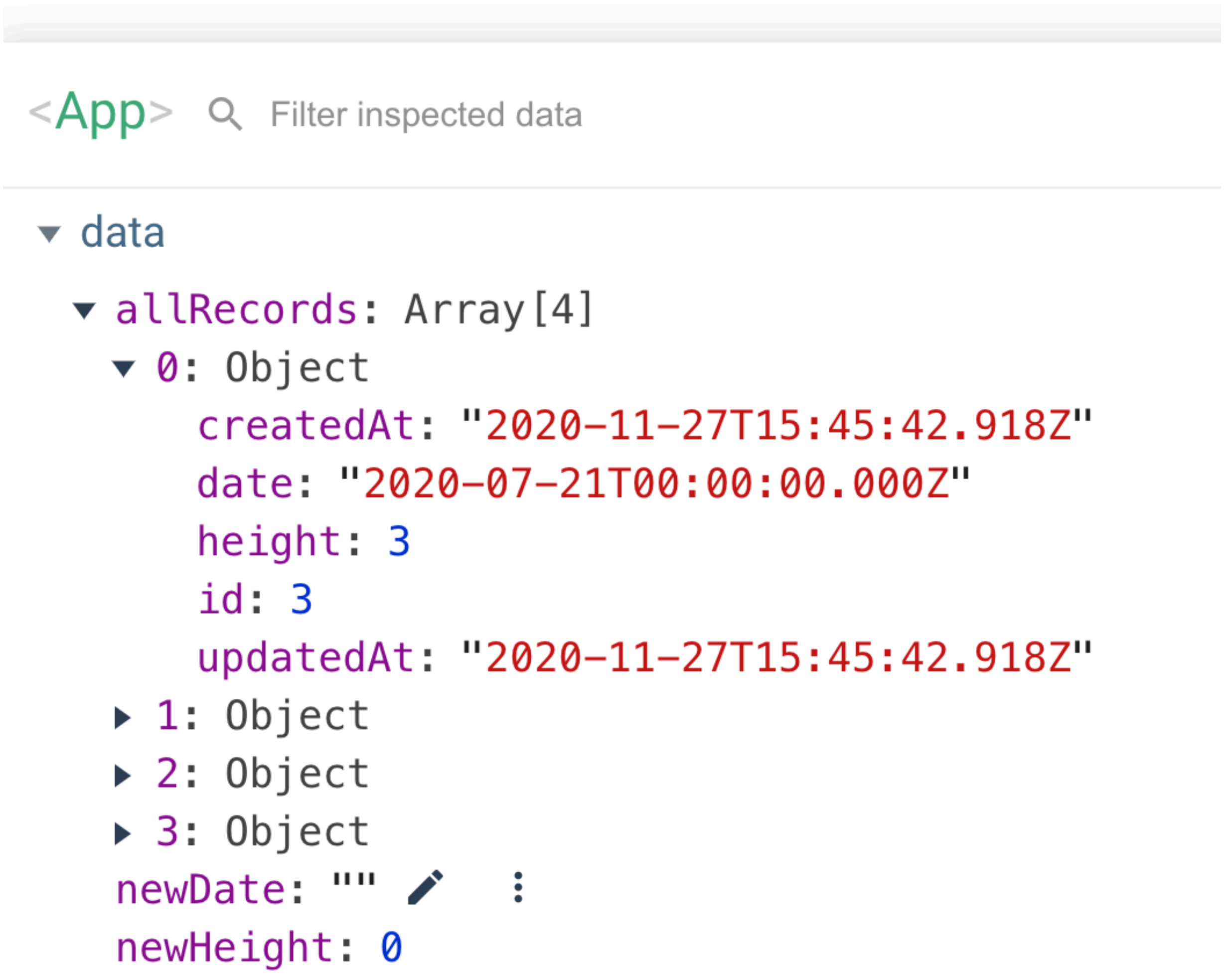
```
<script>
import PlantChart from './components/PlantChart.vue'
```

```
export default {
  name: 'App',
  components: {
    PlantChart
  },
  data() {
    return {
      newHeight: 0,
      newDate: '',
      allRecords: []
    }
  },
  mounted() {
    this.getAllData()
  },
  methods: {
    addRecord() {
      // todo validation
      let record = {height: this.newHeight, date: new
Date(this.newDate)}
      console.log(record)
      this.
$plant_record_api.addPlantRecord(record).then( response => {
        this.getAllData()
      })
    },
    getAllData() {
      this.$plant_record_api.getAllPlantRecords().then(records
=> {
        this.allRecords = records
      })
    }
}
}
</script>
```

Check vue dev tools



```
▼ <Root>
  ▼ <App> = $vm0
      <PlantChart> = $vm1
```



```
<App>   Filter inspected data

▼ data
  ▼ allRecords: Array[4]
    ▼ 0: Object
        createdAt: "2020-11-27T15:45:42.918Z"
        date: "2020-07-21T00:00:00.000Z"
        height: 3
        id: 3
        updatedAt: "2020-11-27T15:45:42.918Z"
    ▶ 1: Object
    ▶ 2: Object
    ▶ 3: Object
    newDate: ""
    newHeight: 0
```

Now to display in chart!

```
<template>
  <div id="app">
```

```
    <label>Height</label>
    <input type="number" v-model="newHeight">
    <br>
    <label>Date</label>
    <input type="date" v-model="newDate">
    <br>
    <button v-on:click="addRecord">Add record</button>

    <plant-chart v-bind:chartData="chartData"/>
  </div>
</template>

<script>
import PlantChart from './components/PlantChart.vue'

export default {
  name: 'App',
  components: {
    PlantChart
  },
  data() {
    return {
      newHeight: 0,
      newDate: '',
      allRecords: [],

    }
  },
  mounted() {
    this.getAllData()
  },
  methods: {
    addRecord() {
      // todo validation
      let record = {height: this.newHeight, date: new
Date(this.newDate)}
```

```
      console.log(record)
      this.
$plant_record_api.addPlantRecord(record).then( response => {
        this.getAllData()
      })
    },
    getAllData() {
      this.$plant_record_api.getAllPlantRecords().then(records
=> {
        this.allRecords = records
      })
    }
  },
  computed: {
    chartData() {
      let labels = this.allRecords.map(rec => rec.date) // all
the dates
      let heights = this.allRecords.map(rec => rec.height) //
all the heights

      return {
        labels: labels,
         datasets: [ {
          label: 'Height for date',
          data: heights
        }]
      }
    }
  }
}
</script>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```

And chart

```
<!-- no template -->

<script>

import { Line, mixins } from 'vue-chartjs'
let { reactiveProp } = mixins

export default {
  extends: Line,
  name: 'PlantChart',
  mixins: [reactiveProp],
  mounted() {
    this.renderChart(this.chartData, this.chartOptions)
  }
}
</script>

<!-- Add "scoped" attribute to limit CSS to this component only
-->
<style scoped>

</style>
```

Chart should be shown, updates when new data added

HOWEVER

the x-axis is wrong - the dates are not spaced correctly

and it's a boring color

App.vue

```
 computed: {
    chartData() {

      let labels = this.allRecords.map(rec => rec.date) // all the dates
      let heights = this.allRecords.map(rec => rec.height ) // all the heights

      return {
        labels: labels,
         datasets: [ {
          label: 'Height for date',
          data: heights,
          borderColor: 'teal',    // your choice - HTML color names or RGB
          fill: false  // optional
        }]
      }
    }
  }
```

```
<!-- no template -->
```

```
<script>
```

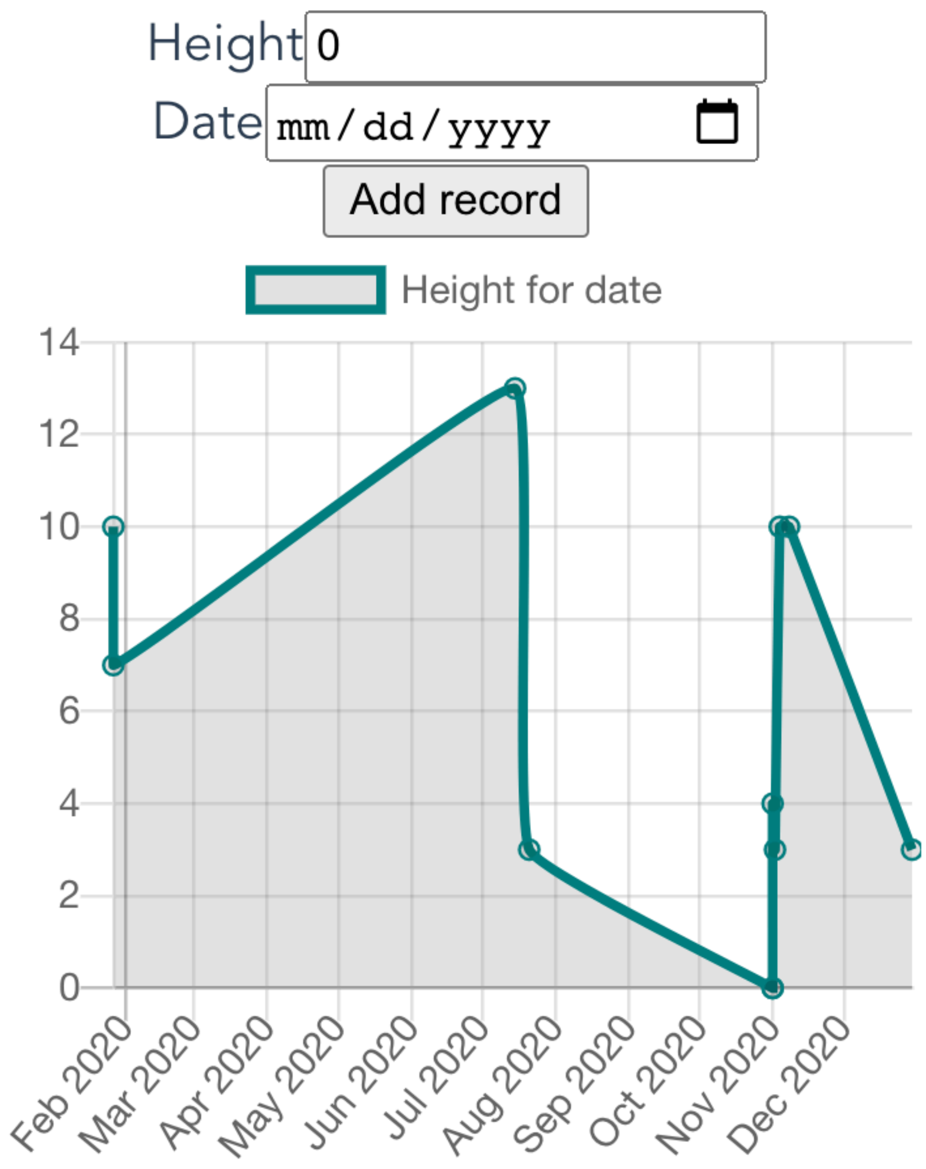
```
import { Line, mixins } from 'vue-chartjs'
let { reactiveProp } = mixins
```

```
export default {
  extends: Line,
  name: 'PlantChart',
  mixins: [ reactiveProp ],
  data() {
    return {
      chartOptions: {
        scales: {
          xAxes: [
            {
              type: 'time',
              distribution: 'linear'  // space out in time
            }
          ]
        }
      }
  }},
  mounted() {
    this.renderChart(this.chartData, this.chartOptions)
  }
}
</script>
```

```
<!-- Add "scoped" attribute to limit CSS to this component only
-->
<style scoped>
```

```
</style>
```

Example Chart:



Height 0

Date mm / dd / yyyy