Get better performance for your agency and ecommerce websites with Cloudways managed hosting. Start with $100, free. →

CONTENTS

// Tutorial //

How To Use Chart.js with Vue.js

Published on January 27, 2018 · Updated on March 12, 2021

Vue.js

By Dave Berning
Developer and Author

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

MANAGE CHOICES

AGREE & PROCEED

Introduction

Chart.js is a powerful way to create clean graphs with the HTML5 `<canvas>` element. With Vue's `data()` object, it is possible to store data and manipulate it to change graphs when needed.

In this article, you will use Chart.js in a sample Vue project to display information about planets in the solar system.

Prerequisites

To complete this tutorial, you will need:

- Node.js installed locally, which you can do by following How to Install Node.js and Create a Local Development Environment.
- Some familiarity with setting up a Vue.js project and using Vue.js components may be beneficial.

This tutorial was verified with Node v15.11.0, `npm` v7.6.1, `vue` v2.6.11, and Chart.js v2.9.4.

Step 1 – Setting Up the Project

To quickly set up the project, this article will recommend using `@vue/cli`.

Note: This article will take the approach of using `npx` to avoid a global installation of `@vue/cli`;

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

```
$ cd vue-async-computed-example
```

Copy

Chart.js can be installed through `npm` with the following command:

```
$ npm install chart.js@2.9.4
```

Copy

At this point, you will have a new Vue project that supports Chart.js.

Step 2 – Creating the Chart Component

This chart will consist of two datasets:

1. The number of moons each planet in the solar system has.
2. The mass of each planet in the solar system.

With these two datasets, we can have different chart types to show correlations in data.

Open your code editor and under `src` directory and `components` subdirectory, create a new `PlanetChart.vue` file.

Every Chart.js chart needs to have a `<canvas>` in the HTML markup. The `id` of the chart is used as a selector to bind the JavaScript to it.

PlanetChart.vue

```
<template>
  <div>
    <canvas id="planet-chart"></canvas>
  </div>
</template>

<script>
import Chart from 'chart.js'

export default {
  name: 'PlanetChart'
}
</script>
```

Copy

Next, you will modify the `App.vue` file to use the new `PlanetChart`:

src/App.vue

```
<template>
```

Copy

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

```
import PlanetChart from './components/PlanetChart.vue'

export default {
  name: 'App',
  components: {
    PlanetChart
  }
}
</script>
```

Save the changes to your file.

In order to keep the component and the configuration separate, you will next be creating a new file for the chart data.

Step 3 – Creating the Chart Data

Creating a chart with Chart.js resembles the following:

```
const ctx = document.getElementById('example');
const exampleChart = new Chart(ctx, {
  type: '',
  data: [],
  options: {},
});
```

Copy

A `<canvas>` element is passed in along with a `type`, `data`, and `options`.

You will be creating a new file that defines these values. Open your code editor and under `src` directory, create a new `planet-data.js` file. Keep in mind, you will want to give it a unique and descriptive name based on the data. You can have several data objects in this file for different charts.

Add the following lines of code to `planet-data.js`:

src/planet-data.js

```
export const planetChartData = {
  type: "line",
  data: {
    labels: ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptu
    datasets: [
      {
        label: "Number of Moons",
        data: [0, 0, 1, 2, 79, 82, 27, 14],
        backgroundColor: "rgba(54,73,93,.5)",
```

Copy

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

```
        backgroundColor: "rgba(71, 183,132,.5)",
        borderColor: "#47b784",
        borderWidth: 3
      }
    ]
  },
  options: {
    responsive: true,
    lineTension: 1,
    scales: {
      yAxes: [
        {
          ticks: {
            beginAtZero: true,
            padding: 25
          }
        }
      ]
    }
  }
};

export default planetChartData;
```

The `type` will be set to `line`. The `data` will consist of of two `datasets`, `backgroundColor`, `borderColor`, and `borderWidth`. The `options` will consist of `responsive`, `lineTension`, and `scales`.

> **Note:** You can reference the Chart.js documentation for more information about line charts, as well as others like `bar`, `polarArea`, `radar`, `pie`, and `doughnut`.

By exporting `planetChartData`, you are allowing that `const` to be imported into another JavaScript file. More importantly, you are separating the data from the component. This best practice allows you to apply changes to only the relevant file and offers greater maintenance over time.

Revisit the `PlanetChart` component and add `planetChartData`:

src/components/PlanetChart.vue

Copy

```
<script>
import Chart from 'chart.js'
import planetChartData from '../planet-data.js'

export default {
  name: 'PlanetChart'
}
</script>
```

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

```
<script>
import Chart from 'chart.js'
import planetChartData from '../planet-data.js'

export default {
  name: 'PlanetChart',
  data() {
    return {
      planetChartData: planetChartData
    }
  }
}
</script>
```

Copy

Note: You can also use ES6 shorthand. Since the data property and value have the same name, you can just use `planetChartData` instead of `planetChartData: planetChartData`.

At this point, Chart.js should be installed and the chart's data should be imported into the `PlanetChart` component.

You should already have a `<canvas>` element created in the component's template. At this point, it's time to initialize the chart and write to the `<canvas>`.

Revisit the `PlanetChart` component and add the create the chart in the `mounted()` lifecycle method:

src/components/PlanetChart.vue

```
<script>
import Chart from 'chart.js'
import planetChartData from '../planet-data.js'

export default {
  name: 'PlanetChart',
  data() {
    return {
      planetChartData: planetChartData
    }
  },
  mounted() {
    const ctx = document.getElementById('planet-chart');
    new Chart(ctx, this.planetChartData);
  }
}
</script>
```

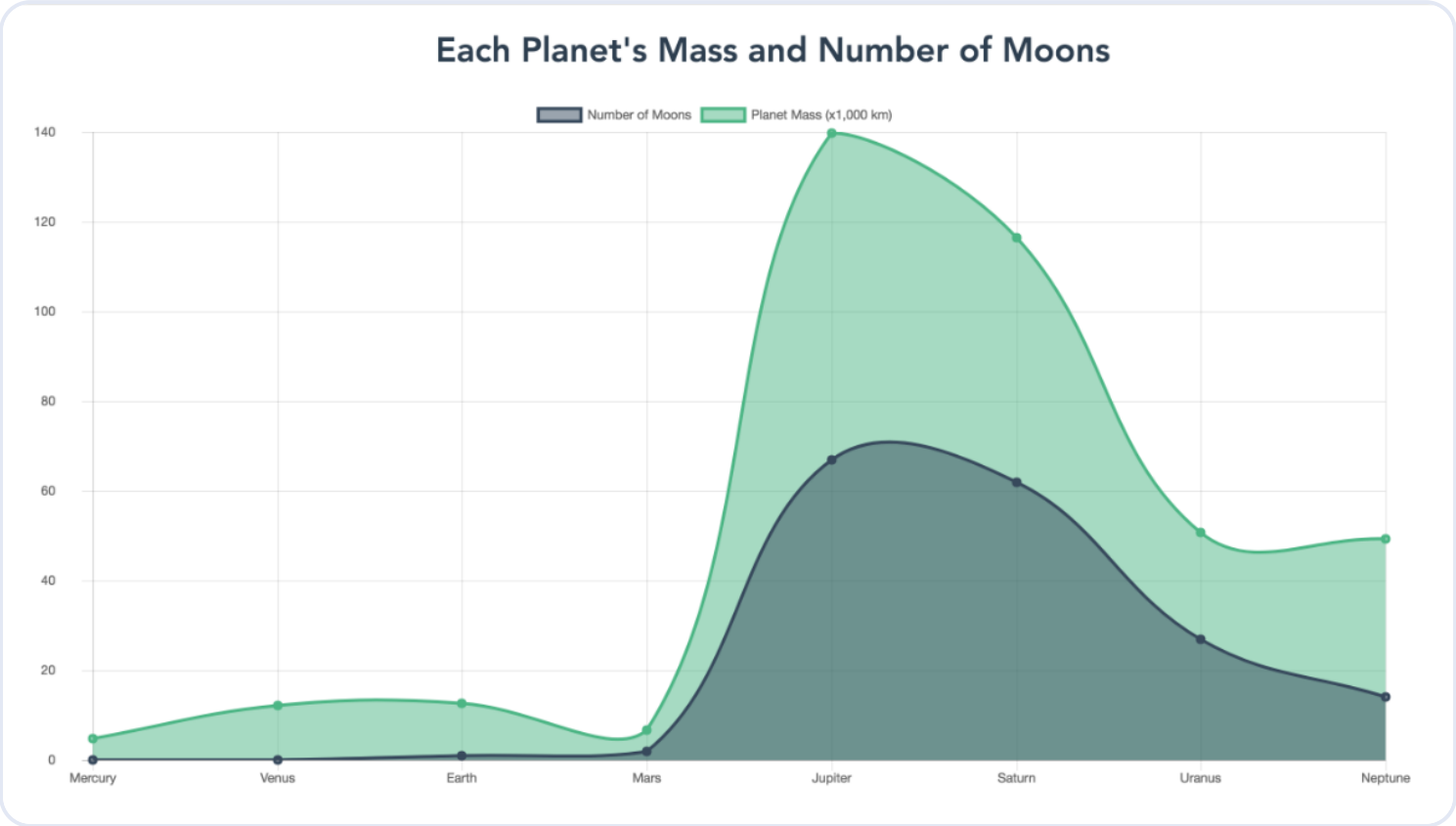
Copy

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

```
$ npm run serve
```

Copy

Open the application in your web browser:



There will be two line charts. One chart displays the number of moons for each planet. THe other chart displays the mass of each planet. The `backgroundColor`, `borderColor`, and `borderWidth` values have affected the appearance of the charts. Interacting with the points on the line will display the `labels`.

Step 4 – Displaying Mixed Charts

Chart.js also supports mixed charts. In this section, you will change the configuration of the charts from line charts to a mix of a line chart for the moon dataset and a bar chart for the mass dataset.

Revisit `planet-data.js` with your code editor. Modify the `type` property of your chart's data and change it to `bar`.

src/planet-data.js

```
export const planetChartData = {
  type: "bar",
  data: { ... },
  options: { ... }
};
```

Copy

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

However, you want to display a mix of bar and line graphs. To change this, in each `dataset` object, add a `type` property under the `label` property. For the first `dataset` object, give it a `type` property with a value of `line` and for the second, give it a `type` property with a value of `bar`:

src/planet-data.js

Copy

```
export const planetChartData = {
  type: "bar",
  data: {
    labels: ["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptu
    datasets: [
      {
        label: "Number of Moons",
        type: "line',
        data: [0, 0, 1, 2, 79, 82, 27, 14],
        backgroundColor: "rgba(54,73,93,.5)",
        borderColor: "#36495d",
        borderWidth: 3
      },
      {
        label: "Planetary Mass (relative to the Sun x 10^-6)",
        type: "bar",
        data: [0.166, 2.081, 3.003, 0.323, 954.792, 285.886, 43.662, 51.514],
        backgroundColor: "rgba(71, 183,132,.5)",
        borderColor: "#47b784",
        borderWidth: 3
      }
    ]
  },
  options: { ... }
};

export default planetChartData;
```

Save the changes to your file.

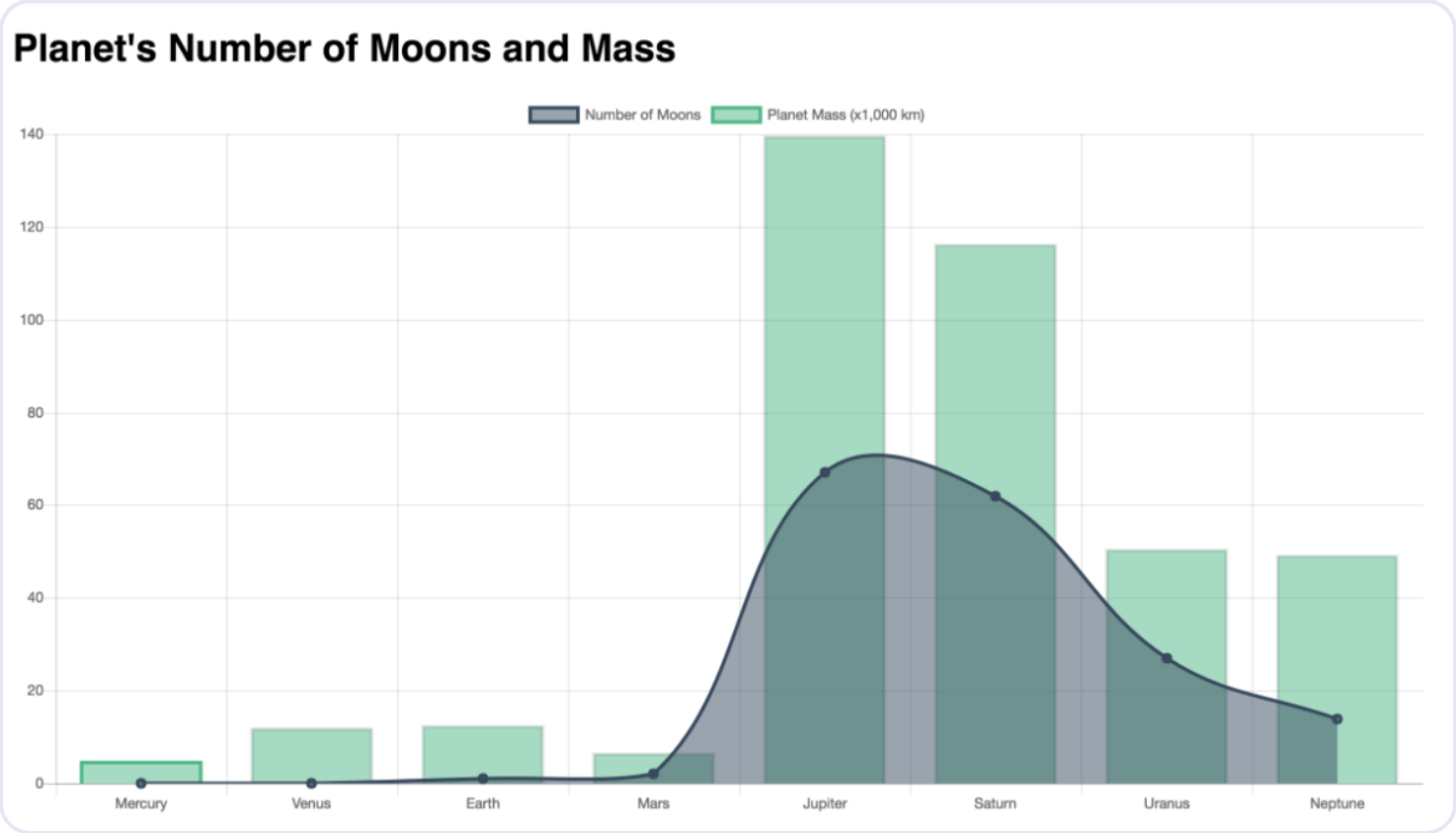
Now, you can run your application with your terminal:

Copy

```
$ npm run serve
```

Open the application in your web browser:

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.



The number of moons for each planet displays as a line chart. The mass of each planet displays as a bar chart.

Conclusion

In this article, you used Chart.js in a sample Vue project to display information about planets in the solar system.

Chart.js can be used for other data visualizations. Explore the samples for inspiration in how it can be incorporated into your projects.

Thanks for learning with the DigitalOcean Community. Check out our offerings for compute, storage, networking, and managed databases.

Learn more about us →

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

Dave Berning Author
Developer and Author

I'm a software engineer from Cincinnati. I work on TypeScript apps with Vue.js. Currently a Senior Front-End Engineer at Enodo, based in Chicago.

Still looking for an answer? Ask a question Search for more help

Was this helpful? Yes No

Comments

3 Comments

Leave a comment ...

This textbox defaults to using `Markdown` to format your answer.

You can type `!ref` in this text area to quickly search our full set of tutorials, documentation & marketplace offerings and insert the link!

Sign In or Sign Up to Comment

Holambro • June 9, 2021

Great walkthrough. Helped me a lot. Just 2 remarks:

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

Reply

RyuZebian • April 5, 2021

First off, thank you for a well written guide!

I've been struggling with a variety of errors for days trying to use either v-charts and vue-echarts with echarts, but today I decided to just try Chart.js. This guide was the clearest and to-the-point of many, but I'm currently stuck after following the steps...

npm packages: ├── @vue/cli@4.5.12 ├── @vue/composition-api@1.0.0-rc.6 ├── chart.js@3.0.2 ├── echarts@5.0.2 ├── vue-chartjs@3.5.1 ├── vue-echarts@6.0.0-rc.4 └── vue@2.6.12

Copy pasting your code, I get the error:

"Error in mounted hook: "TypeError: chart_js__WEBPACK_IMPORTED_MODULE_0__.default is not a constructor"

found in

--> <PlanetChart> at src/components/PlanetChart.vue <App> at src/App.vue <Root>"

Do you have any idea what could've gone wrong? :S

Show replies Reply

hottehead • January 20, 2021

Awesome! Thanks a lot. This works with vue3! Tried vue-chartjs package and it does not support vue3. So this saved me.

Show replies Reply

This work is licensed under a Creative Commons Attribution-NonCommercial- ShareAlike 4.0 International

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.

Try DigitalOcean for free

Click below to sign up and get **$200 of credit** to try our products over 60 days!

Sign up

Popular Topics

Ubuntu

Linux Basics

JavaScript

Python

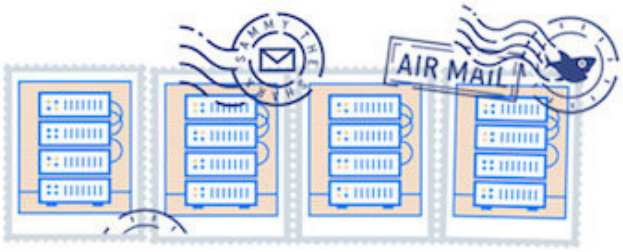
MySQL

Docker

Kubernetes

All tutorials →

Free Managed Hosting →



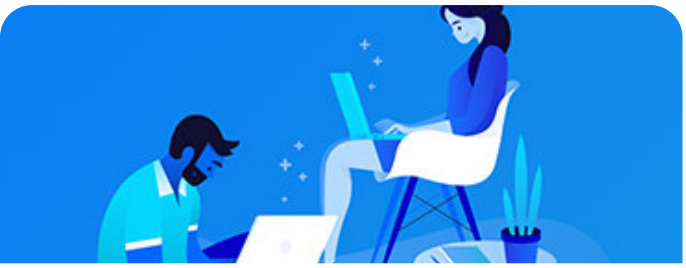
Get our biweekly newsletter

Sign up for Infrastructure as a

Hollie's Hub for Good

Working on improving health and education, reducing inequality, and spurring economic growth? We'd

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.



Become a contributor

You get paid; we donate to tech nonprofits.

Learn more →

Featured on Community

Kubernetes Course · Learn Python 3 · Machine Learning in Python · Getting started with Go · Intro to Kubernetes

DigitalOcean Products

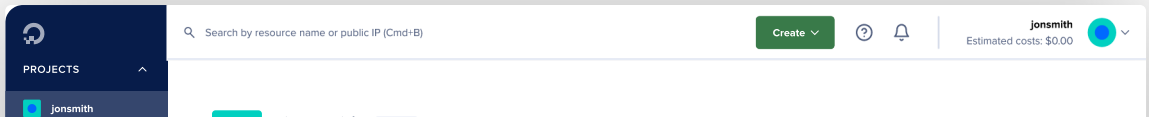
Cloudways · Virtual Machines · Managed Databases · Managed Kubernetes · Block Storage · Object Storage · Marketplace · VPC · Load Balancers

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you're running one virtual machine or ten thousand.

Learn more →

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.



Company

- About
- Leadership
- Blog
- Careers
- Customers
- Partners
- Channel Partners
- Referral Program
- Affiliate Program
- Press
- Legal
- Security
- Investor Relations
- DO Impact

Products

- Products Overview
- Droplets
- Kubernetes
- App Platform
- Functions
- Cloudways
- Managed Databases
- Spaces
- Marketplace
- Load Balancers
- Block Storage
- Tools & Integrations
- API
- Pricing
- Documentation
- Release Notes
- Uptime

Community

- Tutorials
- Q&A
- CSS-Tricks
- Write for DOnations
- Currents Research
- Hatch Startup Program
- deploy by DigitalOcean
- Shop Swag
- Research Program
- Open Source
- Code of Conduct
- Newsletter Signup
- Meetups

Solutions

- Website Hosting
- VPS Hosting
- Web & Mobile Apps
- Game Development
- Streaming
- VPN
- SaaS Platforms
- Cloud Hosting for Blockchain
- Startup Resources

Contact

- Support
- Sales
- Report Abuse
- System Status
- Share your ideas

© 2023 DigitalOcean, LLC. All rights reserved.

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may include site operation, analytics, enhanced user experience, or advertising. You may choose to consent to our use of these technologies, or manage your own preferences.