

Мы переписывали фреймворк 3 раза — и сделаем это снова

ТЯЖЕЛЫЕ УРОКИ VUETIFY

Что такое Vuetify



COLORS



Primary: #2d2c69

MORE

SCHEME & MIXER

Tailwind

Analogous

VARIANT & CONTRAST

Contrast and readability icons

FONTS

Headings: Roboto

Body: Noto Sans

Font size root: 16

- Showcase
- Theme Specification
- Palette Colors
- Export

Daily activities

- New Icon** - Web app favicon - 09:30 am
- Design Stand Up** - Hangouts - 11:00 am
- Lunch break** - 01:15 pm
- Finish Home Screen** - Web app - 03:30 pm
- Gym** - Cross training - 05:00 pm

Inbox

Today

- Brunch this weekend?**
Ali Connors — I'll be in your neighborhood doing errands this...
- Summer BBQ**
to Alex, Scott, Jennifer — Wish I could come, but I'm out of town...
- Oui oui**
Sandra Adams — Do you have Paris recommendations? Have yo...

User Registrations

Last Campaign Performance

last registration 26 minutes ago

OPEN FULL REPORT

My Profile

Johnathan Miller
@johnmiller

324 Photos | 51 Posts | 937 Followers

Top Sales

- Soundspace E45** headphones - 41 x @ 35.00
- Nebula Neptun X3** mobile projectors - 2 x @ 295.00
- NuPhy Field75** wireless keyboards - 4 x @ 172.00
- F-HDMI-01-VS** cables - 33 x @ 9.50

ELEVATED

Headline

Greyhound divisely hello coldly fownderfully

BUTTON

Дисклеймер

Я не буду рассказывать о том, как пользоваться Vuetify

О чём поговорим

01

Контекст

02

Кодовые
ошибки

03

Некодовые
ошибки

04

Шокирующий
анонс

КТО

Андрей Ёлкин

Senior Software Engineer в [Cycloid.io](https://cycloid.io)

Core Team Member [Vuetify](https://vuetify.com)



[@andreyyolkin](https://github.com/andreyyolkin)



[@yolkin](https://www.linkedin.com/in/yolkin)





Делаем бэкенд на Nuxt

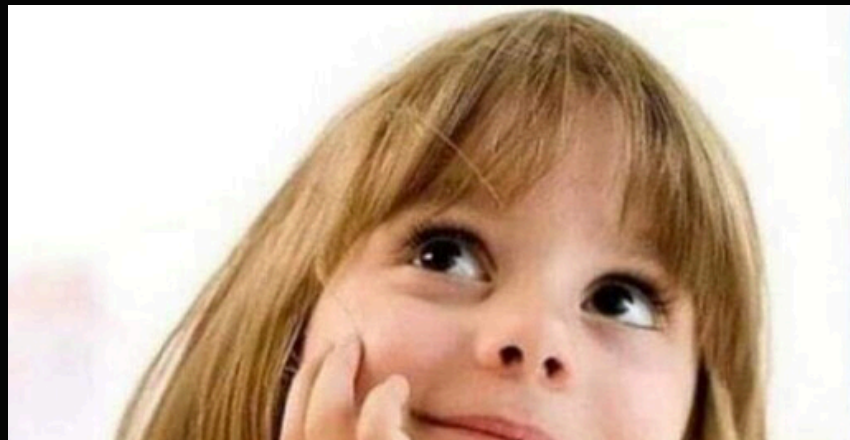


Андрей Ёлкин

Cycloid







Vuetify сегодня

45к+

в Discord

10 лет

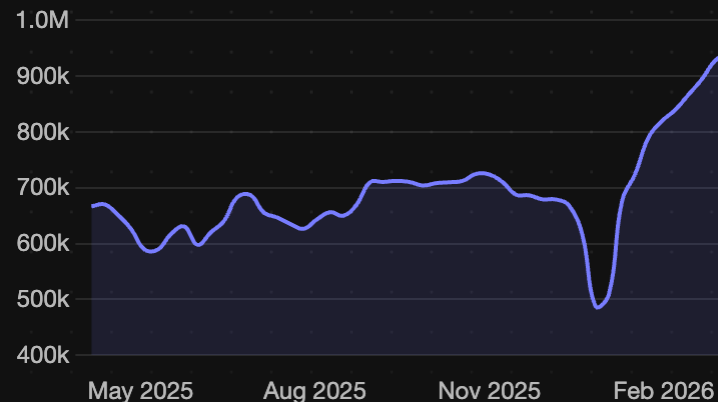
в опенсорсе

80+

КОМПОНЕНТОВ

41к+

звёзд на GitHub



Еженедельные загрузки (npm)

Главный вопрос

Почему мы переписывали всё три раза?

Очевидный ответ: экосистема

UI-библиотека — это баланс параметров

UNSTYLED 21%

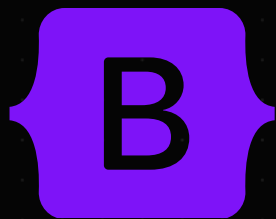
79% OPINIONATED



LOGICLESS 88%

12% FULLY LOGICAL





Bootstrap

Стили и готовые компоненты с ограниченной логикой

UNSTYLED 15%

85% OPINIONATED



LOGICLESS 45%

55% FULLY LOGICAL





Headless UI

Headless primitives: вся логика, нулевой стайлинг.

UNSTYLED 94%

6% OPINIONATED



LOGICLESS 5%

95% FULLY LOGICAL





Vuetify

Material Design + полный набор логики — всё в одной коробке.

UNSTYLED 8%

92% OPINIONATED

LOGICLESS 8%

92% FULLY LOGICAL

Как мы переписывали стили



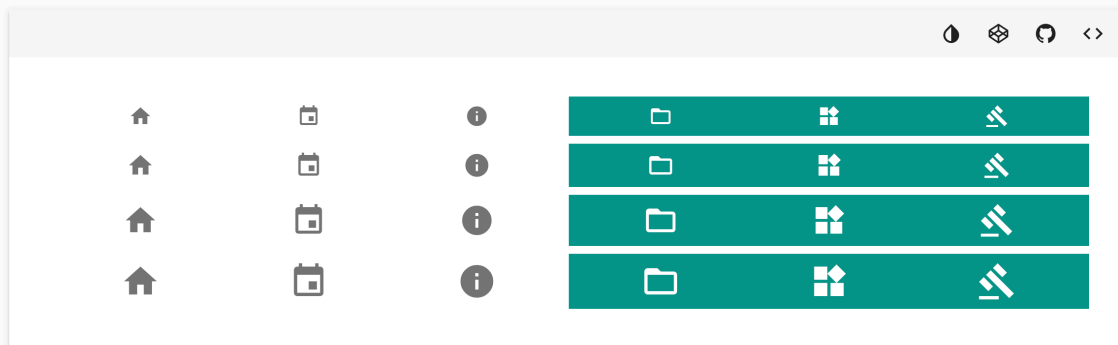
- Getting started
- Framework options
- UI Components**
 - API explorer
 - Browse components
 - Alerts
 - Avatars
 - Badges
 - Bottom navigation
 - Bottom sheets
 - Breadcrumbs
 - Buttons
 - Buttons: Floating action buttons
 - Calendars
 - Cards
 - Carousels
 - Chips
 - Data iterator
 - Data tables
 - Dialogs
 - Dividers
 - Expansion panels
 - Footer
 - Groups

Icon

The `v-icon` component provides a large set of glyphs to provide context to various aspects of your application. Vuetify icons utilize Google's Material Icons font library. For a list of all available icons, visit the official [Material Icons](#) page.

Usage

Icons come in two themes (light and dark), and four different sizes (standard, medium, large, and x-large). If you want to use SVG icons with `VIcon` component, read about using SVG icons with [VIcon](#) [here](#).



API

PROPS		SLOTS	
Name	color	Default	undefined
Type	string		

Icon

Usage

API

Examples

Up Next



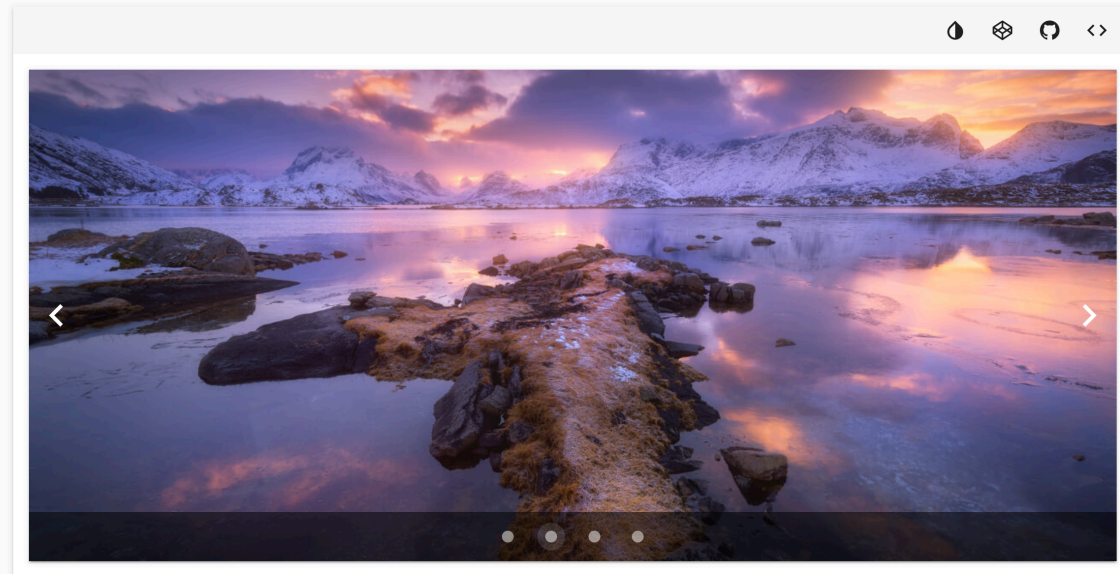
- Getting started
- Framework options
- UI Components
 - API explorer
 - Browse components
 - Alerts
 - Avatars
 - Badges
 - Bottom navigation
 - Bottom sheets
 - Breadcrumbs
 - Buttons
 - Buttons: Floating action buttons
 - Calendars
 - Cards
 - Carousels
 - Chips
 - Data iterator
 - Data tables
 - Dialogs
 - Dividers
 - Expansion panels
 - Footer
 - Groups

Carousel

The `v-carousel` component is used to display large numbers of visual content on a rotating timer.

Usage

A carousel by default has a slide transition.



API

Carousel

Usage

API

Examples

Up Next





Vuetify
Branch:

Your Logo Here

FIND OUT HOW

Core documentation

Vuetify ▾

Layout ▾

Style ▾

Motion ▾

Components ▾

Alerts

Breadcrumbs

Bottom navigation

Buttons

Buttons: Floating Action Buttons

Cards

Carousel

Chips

Data tables

Dialogs

Dividers

Expansion panels

Footer

Grid Lists

Icons

Examples

#1 Flat

Light Theme

NORMAL

PRIMARY

DISABLED

Dark Theme

NORMAL

PRIMARY

DISABLED

#2 Raised

Raised Light Theme

NORMAL

ANOTHER

DISABLED

Raised Dark Theme

NORMAL

ERROR

DISABLED

Vuetify 1: Stylus

Быстрый старт

Material 1.0

```
button-style(  
  border-radius: 2px  
  padding: 6px 16px  
  text-transform: uppercase  
  font-weight: 500  
  
  &.primary  
    background-color: $primary-color  
    color: white
```

Помните Stylus?

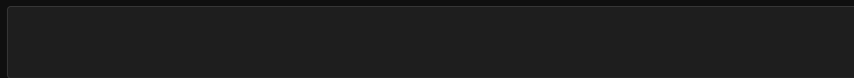
Это он сейчас



- Badges
- Banners
- Bars
- Bottom navigation
- Bottom sheets
- Breadcrumbs
- Buttons**
- Buttons: Floating Action Button
- Calendars
- Cards
- Carousels
- Chips
- Dialogs
- Dividers
- Expansion panels
- Footers
- Form inputs & controls
- Grid system
- Groups
- Hover
- Icons
- Images
- Lazy
- Lists
- Menus

Buttons

The `v-btn` component replaces the standard html button with a material design theme and a multitude of options. Any color helper class can be used to alter the background or text color.



Usage

Buttons in their simplest form contain uppercase text, a slight elevation, hover effect, and a ripple effect on click.

<
DEPRESSED
ICON
OUTLINED
PLAIN
RAISED
>

CLICK ME

Options

- Block
- Disabled
- Large
- Loading
- Small
- X Large
- X Small

Elevation

Color ▼

```

<v-btn
  elevation="2"
  large
  outlined
  >
```

Contents

- Usage
- API
- Caveats
- Examples
- Props
- Block
- Depressed
- Floating
- Icon
- Loaders
- Outlined
- Plain
- Rounded
- Sizing
- Text
- Tile
- Misc
- Raised



- Getting started
- Features
 - Accessibility (a11y)
 - Bidirectionality (LTR/RTL)
 - Display Breakpoints
 - Global configuration
 - Icon Fonts
 - Internationalization (i18n)
 - Layouts
 - Presets
 - SASS variables**
 - Programmatic scrolling
 - Theme
 - Treeshaking
- Styles and animations
- UI Components
- API
- Directives

Variable API

There are many SASS/SCSS variables that can be customized across the entire Vuetify framework. You can browse all the variables using the tool below:

Some color-related variables for components are defined in the global material-theme variables: `$material-light` / `$material-dark`

Search SASS Variables

Example variable file

Below is an example of what a custom variable file could look like:

```
// src/sass/variables.scss

// Globals
$body-font-family: 'Work Sans', serif;
$border-radius-root: 6px;
$font-size-root: 14px;

// Variables must come before the import
$btn-letter-spacing: 0;
$btn-font-weight: 400;
$list-item-title-font-size: 0.929rem;
$list-item-dense-title-font-size: 0.929rem;
$list-item-dense-title-font-weight: initial;
$fab-icon-sizes: (
  'small': 20
);
$btn-sizes: (
  'default': 41,
  'large': 54
);
```

Contents

- Vue CLI install
- Nuxt install
- Webpack install
- Variable API**
- Example variable file
- Usage in templates
 - Global variables
 - Component variables
- Caveats
 - Duplicated CSS
 - Compilation time

Start your next project with









Vuetify 2: Sass + @import

```
@import "variables";
@import "mixins";
@import "buttons";
@import "cards";
@import "dialogs";
@import "material-theme.scss";

$primary-color: #1976d2; // из variables.scss
$primary-color: #2196f3; // из material-theme.scss

.v-btn {
  background-color: $primary-color;
  color: #fff;
}
```



-  Introduction ▼
-  Getting started ▼
-  Features ▼
-  Styles and animations ▼
-  Common concepts ▼
-  Components ▲

All Components

 API Explorer
[Browse Components](#)

Application

CONTAINMENT

[Bottom sheets](#)[Buttons](#)[Cards](#)[Chips](#)[Dialogs](#)[Dividers](#)[Expansion panels](#)[Lists](#)[Menus](#)[Overlays](#)[Sheets](#)[Toolbars](#)[Tooltips](#)

Buttons

The `v-btn` component replaces the standard html button with a material design theme and a multitude of options. Any color helper class can be used to alter the background or text color.

- [Edit this page](#)
 - [Figma Design](#)
 - [Report a Bug](#)
 - [Open issues](#)
 - [View on GitHub](#)
 - [Design Specification](#)
- [Copy Page as Markdown](#)

Usage

Buttons in their simplest form contain uppercase text, a slight elevation, hover effect, and a ripple effect on click.

Default
Outlined
Tonal
Text
Plain

▼
🔍
📄
▲

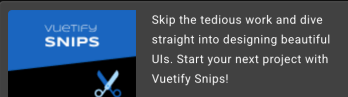
▼ BUTTON

Configuration

- Icon
- Prepend icon
- Append icon
- Stacked

```
<v-btn prepend-icon="$vuetify" variant="tonal">
  Button
</v-btn>
```

Contents

[Usage](#)[API](#)[Anatomy](#)[Guide](#)[Props](#)[Density](#)[Size](#)[Block](#)[Rounded](#)[Elevation](#)[Ripple](#)[Variants](#)[Icon](#)[Loaders](#)[Inside of bars](#)[Slots](#)[Icon color](#)[Spaced](#)[Custom loader](#)[Examples](#)[Discord event](#)[Survey group](#)[Tax form confirmation](#)[Dialog action](#)[Cookie settings](#)[Readonly buttons](#)[Global Configuration](#)[Aliasing](#)[SASS Variables](#)[Defaults Side Effects](#)[Banners](#)[Bottom navigation](#)



- Introduction
- Getting started
- Features
- Styles and animations
- Common concepts
- Components
- All Components
- API Explorer
Browse Components
- Application

CONTAINMENT

Bottom sheets

Buttons

Cards

Chips

Dialogs

Dividers

Expansion panels

Lists

Menus

Overlays

Sheets

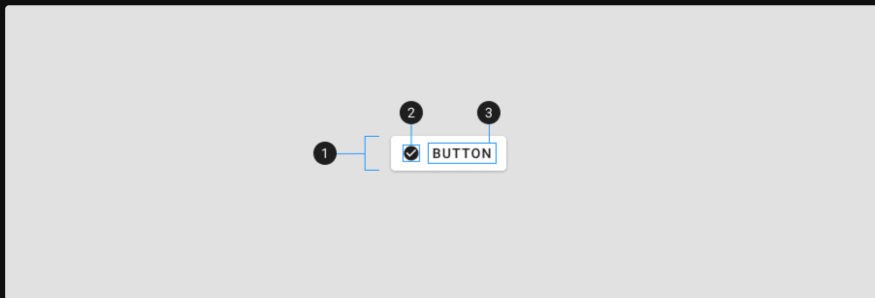
Toolbars

Tooltips

Anatomy

The recommended placement of elements inside of `v-btn` is:

- Place text in the center
- Place visual content around container text



Element / Area	Description
1. Container	In addition to text, the Button container typically holds a <code>v-icon</code> component
2. Icon (optional)	Leading media content intended to improve visual context
3. Text	A content area for displaying text and other inline elements

Guide

The `v-btn` component is commonly used throughout Vuetify and is a staple for any application. It is used for everything from navigation to form submission; and can be styled in a multitude of ways.

The following code snippet is an example of a basic `v-btn` component only containing text:

```
<v-btn>Button</v-btn>
```

HTML

Slots

Icon color

Spaced

Custom loader

Examples

Discord event

Survey group

Tax form confirmation

Dialog action

Cookie settings

Readonly buttons

Global Configuration

Aliasing

SASS Variables

Defaults Side Effects

Banners

Bottom navigation

Button groups

Cards

Snackbars

Toolbars and AppBars

Accessibility

ARIA Attributes

Keyboard Navigation

Accessible Labels

Touch Target Size

Sponsors



Support



Vuetify 3: Hybrid Sass

```
@use 'vuetify/settings' with (  
  $utilities: false,  
  $body-font-family: ('Remora Sans', sans-serif),  
);
```

```
:root {  
  --v-theme-primary: #1976d2;  
}  
  
.v-btn {  
  background-color: var(--v-theme-primary);  
  color: white;  
}
```

Vuetify 4: ничего важного

```
@layer vuetify-utilities {
  @layer theme-base {
    :root {
      --v-theme-primary: #1976d2;
    }
  }
}

@layer vuetify-components {
  .v-btn {
    background-color: var(--v-theme-primary);
    color: white;
  }
}
```

Итого

Версия	Подход к стилям
Vuetify 1	Stylus
Vuetify 2	Sass
Vuetify 3	Sass + CSS Variables
Vuetify 4	Sass + CSS Layers (+ Variables)

Как мы переписывали логику

v1/v2: Vue.extend

```
const baseMixins = mixins(
  VSheet, Routable, Positionable, Sizeable,
  GroupableFactory('btnToggle'), ToggleableFactory('inputValue')
)

export default baseMixins.extend<options>().extend({
  name: 'v-btn',
  render (h): VNode {
    const children = [this.genContent(), this.loading && this.genLoader()]
    const { tag, data } = this.generateRouteLink()
    const setColor = this.hasBg ? this.setBackgroundColor : this.setTextColor

    return h(tag, this.disabled ? data : setColor(this.color, data), children)
  }
})
```

v1/v2: Vue.extend

Да

Практичный выбор для Vue 2 + TS
Render-функции давали контроль

Но

Миксины и фабрики увеличивали
скрытый контекст
\$emit и слоты почти не
типизировались
Порог входа для контрибьюторов рос

Vuetify 3: Эпоха TSX

```
export const VBtn = genericComponent<VBtnSlots>(()({
  name: 'VBtn',
  props: makeVBtnProps(),
  setup (props, { slots }) {
    const { themeClasses } = provideTheme(props)
    const { variantClasses } = useVariant(props)
    useRender(() => (
      <props.tag
        class={[
          'v-btn',
          themeClasses.value,
          variantClasses.value,
        ]}
      >
        { slots.default?.() }
      </props.tag>
    ))
  }
})
```

Vuetify 3: Эпоха TSX

Да

Полная проверка типов пропсов и событий

Легкая композиция через композаблы

Ошибки подсвечиваются в IDE, а не в рантайме

Но

Цена за сложную типизацию выросла

DX для контрибьюторов местами просел

...и ещё кое-что

Как работает TSX во Vue

- Runtime-ориентирован: VNode-дерево строится при каждом рендере
- Компилируется через `@vue/babel-plugin-jsx`
- Не получает шаблонных оптимизаций — patch flags, hoisting, block tree

@vue/babel-plugin-jsx

Нежеланный ребёнок экосистемы

Цена компромисса

```
export function ensureValidVNode (vnodes: VNodeArrayChildren) {
  return vnodes.some(child => {
    if (!isVNode(child)) return true
    if (child.type === Comment) return false
    return child.type !== Fragment ||
      ensureValidVNode(child.children as VNodeArrayChildren)
  }) ? vnodes : null
}
```

Эта функция нужна, чтобы не рендерить обёртку вокруг пустого слота. Она работает за счёт обхода VNode-дерева — `isVNode`, `Comment`, `Fragment`, рекурсия в `children`.

"Ho" №2

Больше нет причин писать на TSX

Vue Vapor

ПЭЙН, Я НЕ ЧУВСТВУЮ VNODE

КТО

Андрей Ёлкин

Vue-энтузиаст

Заслуженный член MSK VueJS

Республики Польша



[@andreyyolkin](#)



[@yolkin](#)



```
<script setup>
import { ref } from 'vue'
const msg = ref('Hello World!')
</script>

<template>
  <h1>{{ msg }}</h1>
  <input v-model="msg" />
</template>
```

```
<script setup>
import { ref } from 'vue'
const msg = ref('Hello World!')
</script>

<template>
  <h1>{{ msg }}</h1>
  <input v-model="msg" />
</template>
```

```
const __sfc__ = {
  setup() {
    const msg = ref('Hello World!')

    const render = () => {
      return createVNodeTree([
        h1(text(msg.value)),
        input(bindModel(msg))
      ])
    }

    return render
  }
}
```

Virtual DOM не бесплатный

- Создание и сравнение деревьев VDOM требует памяти и CPU.
- В масштабных приложениях с тысячами компонентов VDOM становится узким местом.
- Рантайм для поддержки VDOM добавляет лишние килобайты к итоговому приложению.

Vue Vapor

- Код превращается напрямую в DOM-операции (как в SolidJS).
- **Нет накладных расходов** на создание и сравнение деревьев.
- Обновляются только те узлы, которые реально изменились.

-65%

размер бандла

1.66x

скорость diff списка

-42%

пик памяти

```
<script setup vapor>
import { ref } from 'vue'
const msg = ref('Hello World!')
</script>
```

```
<template>
  <h1>{{ msg }}</h1>
  <input v-model="msg" />
</template>
```

```
const h1Template = template("<h1> ")
const inputTemplate = template("<input>")
```

```
const __sfc__ = {
  setup() {
    const msg = ref('Hello World!')
    const h1El = h1Template()
    const inputEl = inputTemplate()
    const h1Text = textNode(h1El)

    bindTextModel(inputEl, msg)
    effect(() => setText(h1Text, msg.value))

    return [h1El, inputEl]
  }
}
```

Как подготовиться к Vapor уже сегодня

1. Полный переход на Composition API

Если вы всё ещё используете Options API, пора начинать миграцию.

2. Отказ от прямого манипулирования VNode

Любые хитрости с `h()`, `render()`-функциями или прямым созданием VNodes не будут работать в Vapor-компонентах.

3. Тестирование проекта на бета-версии

Можно поставить бета-версию Vue 3.6 и попробовать `<script setup vapor>` или использовать `createVaporApp`

Вопросы?

Попридержите



Денис Чернов 

до вапора дожить надо

16:37

Что это значит для Vuetify 5

Внутренние утилиты больше не работают

В Vapor нет VNode'ов

```
if (!isVNode(child)) return true // всегда true
if (child.type === Comment) return false // никогда не сработает
```

В Vapor-режиме `ensureValidVNode` тихо превращается в no-op — всегда возвращает контент как "валидный", даже если внутри пусто.

Vuetify уже от этого уходит

PR [#22483](#), ветка `next`, в драфте.

Паттерн миграции один на всю кодовую базу:

```
slots.foo?.() ?? fallback
// →
renderSlot(slots, 'foo', () => fallback)
```

Спасёт ли нас vue-jsx-vapor?

- Преемник babel-jsx, на Rust (Охс), перф уровня Vapor
- Поддерживает и VDOM, и Vapor-выход
- Vuetify уже пробует мигрировать на него



zhiyuanzmj

Vue & Vue Macros core team member. vue-jsx-vapor & ts-macro author

JSX в Vue структурно на отшибе

Сначала был недоприоритизированный `babel-jsx` под крышей core. Теперь — быстрый `vue-jsx-vapor`, но на одном человеке.

Каждый раз, когда мы выбираем TSX, мы ставим на инструмент, в который Vue стратегически не инвестирует.

Получается Roadmap

- Переход на vue-jsx-варог
- Переход на SFC и отказ от VNode
- ???
- PROFIT!!!

Суровая реальность OSS

Это не только файлы с кодом

Core-team в 2017



John Leider

@johnleider

Core



Kael Watts-Deuchar

@KaeLWD

Core

Завышенные амбиции

- Переоценили скорость перехода на Vue 3
- Недооценили сложность миграции
- Ожидания сообщества разошлись с реальностью



John Leider 

@zeroskillz

Anyone looking for a Senior FE Dev?



Проблема не в твите

Проблема в непрозрачности коммуникации



John Leider ✓
@zeroskillz

Vuetify isn't going anywhere. Simply put, it's not paying the bills at the moment.



John Leider ✓
@zeroskillz

[@ThomasKientz](#) [@mubaidr](#) It hasn't been added yet. This is mostly due to the fact that no one on the team uses NUXT and we haven't had a community member step out to provide some help in implementing it.



John Leider ✓
@zeroskillz

That's functionality, not a design. Yes, I know tree-view is missing and I don't see anyone lining up to help lol

Выгорание

- Длинные циклы и высокий контекст
- Перегруз ключевых людей
- Замедление темпа развития



daniluk4000 8 янв 2023 в 12:52

Взлет и падение Vuetify. Некролог



8 мин



25K

**Сильная инженерия без
сильной коммуникации
не масштабируется**

Помните Material Design?

Это он сейчас



Vueify MD

- Dashboard
- User Profile
- Regular Tables
- Typography
- Icons
- Google Maps
- Notifications

Upgrade To PRO

Dashboard

Search

Website Views
Last Campaign Performance

Unlinked 10 minutes ago

Daily Sales
+5% increase in today's sales

Unlinked 4 minutes ago

Completed Tasks
Last Last Campaign Performance

Unlinked 30 minutes ago

Followers: **+245**

Last Updated

Website Views: **75,521**

Tracked from Google Analytics

Revenue: **\$34,245**

Last 24 Hours

Bookings: **184**

See More Stats

Employees Stats
New employees on 15th September, 2016

ID	Name	Salary	Country	City
1	Dakota Rice	\$35,738	Niger	Oud-Turnhout
2	Mirna Hooper	\$23,738	Curacao	Smith-Wass
3	Sage Rodriguez	\$56,142	Netherlands	Overland Park

Tasks

- Sign contract for 'What are conference organizers afraid of?' ✕
- Lines From Great Russian Literature? Or E-mails From My Boss? ✕
- Flooded: One year later, assessing what was lost and what was found when a devastating hail storm hit my hometown ✕

Sheldon Admin
Vue + ShardsUI

Overview Customers Products Settings

Search

Download

General

- Dashboard
- Tasks
- Apps
- Chats
- Users
- Secured by Clerk
- Pages
- Auth
- Errors
- Other
- Settings
- Help Center

Dashboard

Overview Analytics Reports Notifications

Total Revenue
\$45,231.89
+20.1% from last month

Subscriptions
+2350
+163% from last month

Sales
+12,234
+19% from last month

Active Now
+573
+201 since last hour

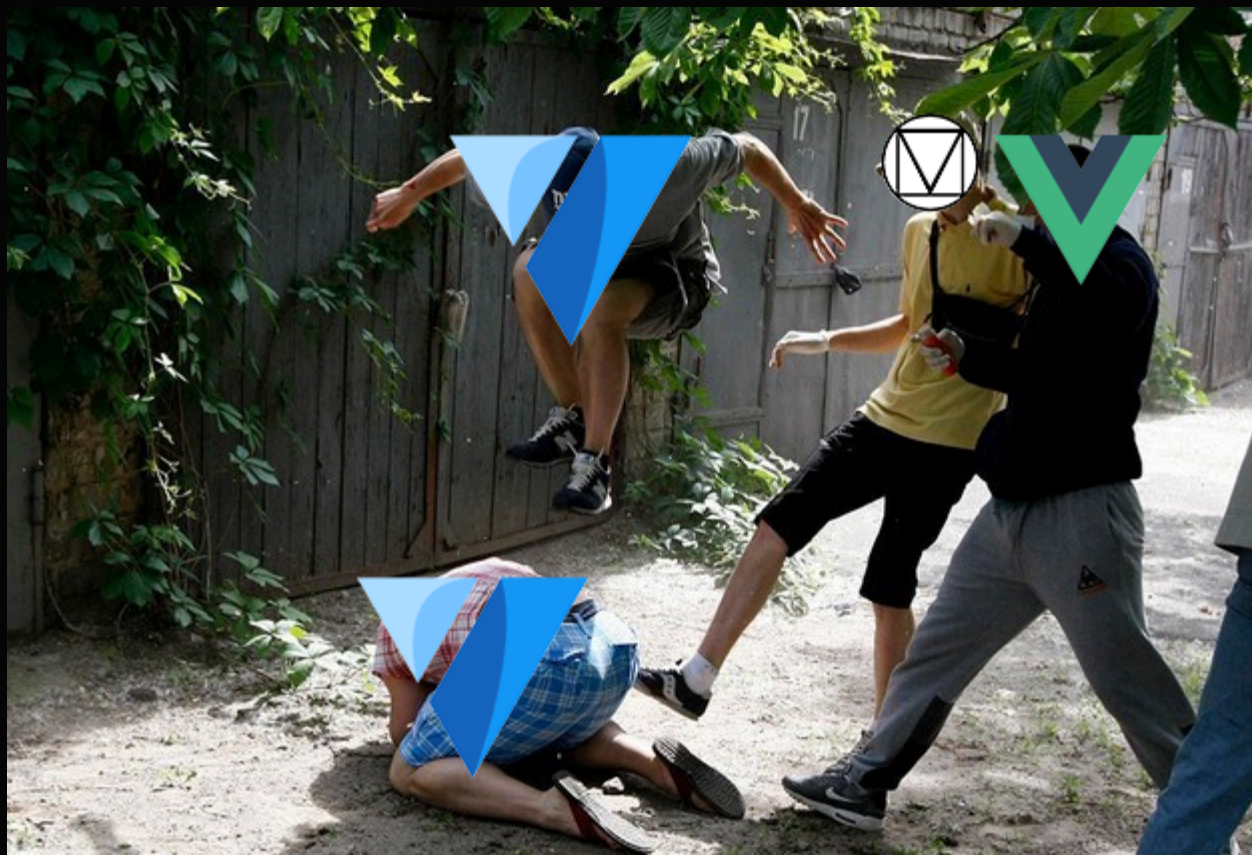
Overview

Month	Revenue
Jan	\$3200
Feb	\$4800
Mar	\$3500
Apr	\$2200
May	\$4700
Jun	\$4500
Jul	\$4600
Aug	\$4800
Sep	\$2800
Oct	\$2000
Nov	\$4500
Dec	\$4300

Recent Sales
You made 265 sales this month.

Avatar	Name	Email	Amount
OM	Olivia Martin	olivia.martin@email.com	+\$1,090.00
JL	Jackson Lee	jackson.lee@email.com	+\$39.00
IN	Isabella Nguyen	isabella.nguyen@email.com	+\$239.00
WK	William Kim	willk@email.com	+\$90.00
SD	Sofia Davis	sofia.davis@email.com	+\$39.00

watchdemo andrew@gmail.com



Но выход есть

Success formula

Project popularity =

Your popularity +

Promotion +

Benefits for users +

Luck



Тренд 1: Экосистема



Тренд 2: Dogfooding

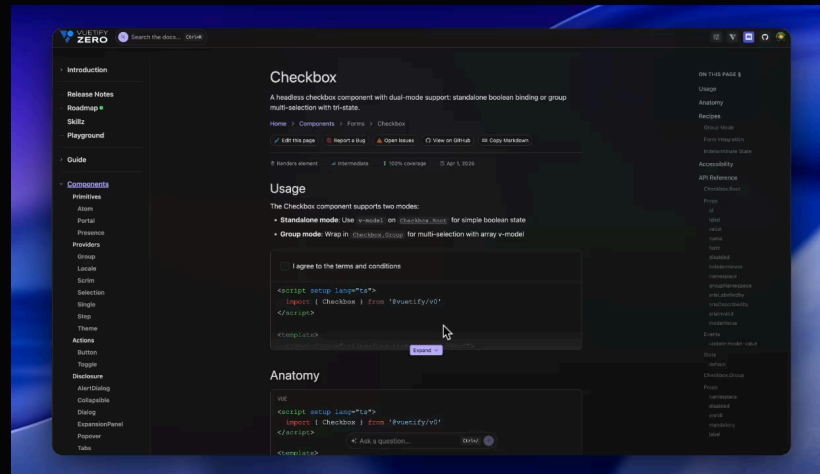
- Валидируем идеи в своих же проектах
- Удаляем бесполезные абстракции
- Тестируем взаимодействие с другими библиотеками

Тренд 3: AI-native

llms.txt

MCP

Ask AI



Тренд 4: Headless

ДО

МОНОЛИТНЫЙ UI-ФРЕЙМВОРК

Логика и стили связаны

Дизайн навязывается библиотекой

Сложнее переиспользовать ядро

ПОСЛЕ

HEADLESS UI-ФРЕЙМВОРК

Логика и стили отдельные

Дизайн полностью отдельно

Ядро минимально



Елизавета Воробьева

Возвращаемся к Vuetify 5

Да, мы опять переписываем

Но не повторяя прошлых итераций — кое-что совершенно новое.

Встречайте — **Vuetyfy 0.**

Vuetify 0

HEADLESS

Контракты a11y, state, keyboard, composition. Стили и DOM-обёрток нет.

STANDALONE

Работает как самостоятельный слой: можно подключать и использовать вне Vuetify.

AI-NATIVE

API и структура оптимизированы под генерацию, анализ и быструю сборку интерфейсов с ИИ.

Color Studio



A theme explorer using `createTheme` with a shared palette of token aliases. Switch between predefined themes, inspect resolved colors, and register new themes at runtime.

- ocean
- forest
- sunset
- midnight
- Cycle



Alex Chen
alex@example.com

Online

Sprint progress

68%



- Design
- Frontend
- Q2



Responsive Layout Detection



Live viewport dimensions, active breakpoint name, and mobile/desktop flags — confirms JS and CSS breakpoints align

lg

1375 x 1003

● Desktop

xs
0px

sm
600px

md
840px

lg
1145px

xl
1545px

xxl
2138px

● xs

false

● sm

false

● md

false

● lg

true

● xl

false

● xxl

false

Scroll Throttle



A scrollable container that tracks position, percentage, and update count — demonstrating RAF throttling of rapid scroll events

Scroll: 0px

Progress: 0%

RAF updates: 0

Item 1

Scroll to see RAF throttling in action

Item 2

Contact Form



Multi-field form with `createForm` integration, lazy validation, and server-side error injection. Each `Form` `Field` creates its own `createElement` and auto-registers with the parent form.

Jane Smith

jane@example.com

Your message...



Hierarchical Tree



A categorized tree with expand/collapse and multi-select checkboxes.

- ✓ Fruits
 - Apple
 - Banana
 - Orange
- ✓ Vegetables
 - Carrot
 - Broccoli
- Bread



- AlertDialog
- Collapsible
- Dialog
- ExpansionPanel
- Popover
- Tabs
- Treeview

Forms

- Checkbox
- Combobox
- Form
- Input**
- NumberField
- Radio
- Rating
- Select
- Slider
- Switch

Semantic

- Avatar
- Breadcrumbs
- Carousel
- Image
- Overflow
- Pagination
- Progress
- Snackbar
- Splitter

> Composables

> API

Input

Headless text input with integrated validation and automatic form registration.

Home > Components > Forms > Input

- Edit this page
- Report a Bug
- Open issues
- View on GitHub
- Copy Markdown

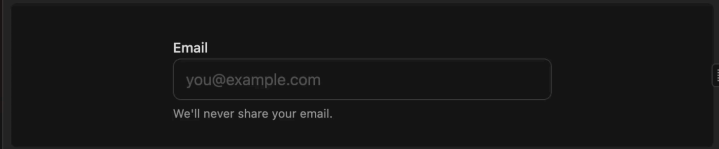
Renders element Intermediate 100% coverage Apr 5, 2026

Usage

The Input supports text, email, password, and other native input types. Validation rules run on blur by default, with `lazy` and `eager` modifiers available.

Basic Input with Validation

Email input with validation rules, inline error display, and helper text.



<> vue

Anatomy

```

VUE
<script setup lang="ts">
  import { Input } from '@vuetify/v0'
</script>

<template>
  <Input.Root>
    <Input.Control>
  </Input.Root>
</template>

```

ON THIS PAGE §

Usage

Basic Input with Validation

Anatomy

Architecture

Examples

Contact Form

Live Search

Recipes

validateOn Modes

Manual Error State

Data Attributes

Accessibility

ARIA Attributes

Keyboard Navigation

API Reference

Input.Root

Props

id

label

name

type

form

disabled

readonly

required

rules

validateOn

error

errorMessagees

namespace

modelValue

focused

Events

Trinity-компонент

CONTEXT

`provide` / `inject` : контракт
между родителем и детьми



COMPOSABLE

Реактивная логика, не
привязанная к DOM



COMPONENT

Оptionальный UI-слой со slot-
props

Context

```
const [useContext, provideContext] = createContext<UserContext>('user')

const user = ref<User>({ id: '123', name: 'John' })
const context: UserContext = { user, update: u => user.value = u }

export const [useUser, provideUser, userCtx] =
  createTrinity<UserContext>(useContext, provideContext, context)

userCtx.update({ id: '456', name: 'Jane' })
```

Composable

```
import { createSelection } from '@vuetify/v0'

const selection = createSelection({
  multiple: true,
})

selection.onboard([
  { id: 1, value: 'Option A' },
  { id: 2, value: 'Option B' },
])
```

TABS

CHIPS

LISTBOX

СВОЙ UI

Component

```
<script setup lang="ts">
  import { Selection } from '@vuetify/v0'
  const selected = shallowRef([])
</script>

<template>
  <Selection.Root v-model="selected" multiple>
    <Selection.Item value="apple">Apple</Selection.Item>
    <Selection.Item value="banana">Banana</Selection.Item>
  </Selection.Root>
</template>
```

Adapter-based plugins

@VUETIFY/V0

0 runtime dependencies

+ АДАПТЕРЫ

logger

i18n

storage

...

Roadmap до v1.0

МАЙ 2026

Alpha



Открыто для фидбека, багов, контрибьютов. API в основном стабильное.

ИЮНЬ 2026

Beta



API замораживается. Финальный круг проверок и фиксов.

ИЮЛЬ 2026

v1.0



Стабильный релиз. API залочены.

8 draft · 96 preview · 19 stable · из 123

0.vuetifyjs.com/roadmap

Основной Vuetify не заброшен

- Расширение core-команды
- Ротация ролей
- Более устойчивая нагрузка

Текущая Core Team



John Leider

@johnleider

Vuetify 0



Kael Watts-Deuchar

@KaeLWD

Core



Jacek Czarniecki

@J-Sek

Core



Henry Aviles

@Haviles04

Ecosystem



Ishan Subedi

@ikushum

Ecosystem



Andrei Elkin

@AndreyYolkin

Tooling

А будет ли Vuetify 6?

V0

Чистая логика и ARIA.

VUETIFY

Готовый Material из коробки.

Vuetify Paper

Три слоя

@VUETIFY/V0

Headless composables, ARIA, поведение

собран на ↓

@VUETIFY/PAPER

V0Paper и контракт стайлинга

собран на ↓

@PAPER/EMERALD · @PAPER/ONYX

Дизайн-системы: токены + styled-компоненты

Контракт: Props → utility classes → data-state CSS

```
<V0Paper
  as="button"
  :rounded="2"
  :elevation="1"
  :data-disabled="disabled || undefined"
  class="emerald-button"
/>
```

PROPS

Декларативный API стайлинга

CLASSES

Массив utility-классов

DATA-*

Стабильный CSS-селектор для состояний

Vuetify → Paper

VUETIFY 3

```
useRounded(props)  
// :style={ '--v-rounded': '8px' }
```

PAPER

```
useRounded(props)  
// :class=['rounded-md']
```

Статус на сегодня

- **Emerald** — PR [#167](#) в `vueifyjs/0`, статус Draft. ~28 компонентов
- Палитра и типографика из Figma-спеки от Juan: шрифт Manrope, иконки Phosphor
- Релиз — после v1.0 v0 (июль-август 2026), вместе с самим Paper

Почему мы уверены — и почему нужен ваш фидбек

Фундамент

ПОЧЕМУ УВЕРЕНЫ

Фокус

Зрелая команда

Проверка практикой

Фидбек

ЧТО НУЖНО

API

DX

Use-case

ИТОГИ

Open Source

- Хорошо, когда платят
- Плохо, когда не платят
- Интересно, когда не платят, а требуют как будто платят

Нюанс

- Хорошо, когда ты Василий Иванович
- Плохо, когда ты Петька
- Интересно, когда Петька уже знал, что такое нюанс

Рабочий проект

- Хорошо, когда на Vue,
- Плохо, когда на React
- Интересно, когда на Laravel

Vuetify

- Хорошо? Конечно хорошо
- Плохо? Да нет же, хорошо
- Интересно? Конечно интересно!

**Мы переписываем не из любви к переписываниям
Мы адаптируемся к реальности**

Спасибо!

