**Инструментарий**

# - Однофайловые компоненты

HTML , CSS, JS - файлдар бір бетте орналасуы

## [Введение](https://ru.vuejs.org/v2/guide/single-file-components.html#%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5)

Негізде компонент глобалный файльда осылай шақырылады, бірақ мұның кемшіліктері бар

Vue.component, с последующим new Vue({ el: '#container' })

Оны реттеу үшін мына тәсілді реттеуге төмендегі тәсіл қолданылады



### [Что насчёт разделения ответственности?](https://ru.vuejs.org/v2/guide/single-file-components.html#%D0%A7%D1%82%D0%BE-%D0%BD%D0%B0%D1%81%D1%87%D1%91%D1%82-%D1%80%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F-%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D0%B8)

HTML , CSS, JS - файлдар бір бетте орналасуы сізге ұнамаса, оны скрипт арқылы жеке жеке шақыруға болады

<template>

<div>Это будет предварительно скомпилировано</div>

</template>

<script src="./my-component.js"></script>

<style src="./my-component.css"></style>

## [Начало работы](https://ru.vuejs.org/v2/guide/single-file-components.html#%D0%9D%D0%B0%D1%87%D0%B0%D0%BB%D0%BE-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D1%8B)

### #[Песочница с примером](https://ru.vuejs.org/v2/guide/single-file-components.html#%D0%9F%D0%B5%D1%81%D0%BE%D1%87%D0%BD%D0%B8%D1%86%D0%B0-%D1%81-%D0%BF%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D0%BE%D0%BC)

### Бір файлдық компоненттер примерімен жұмыс

### #[Для новичков в модульных системах сборки JavaScript](https://ru.vuejs.org/v2/guide/single-file-components.html#%D0%94%D0%BB%D1%8F-%D0%BD%D0%BE%D0%B2%D0%B8%D1%87%D0%BA%D0%BE%D0%B2-%D0%B2-%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8C%D0%BD%D1%8B%D1%85-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%D1%85-%D1%81%D0%B1%D0%BE%D1%80%D0%BA%D0%B8-JavaScript) **ES2015/16** бойынша жұмыс істеп үйрен

### #[Для продвинутых пользователей](https://ru.vuejs.org/v2/guide/single-file-components.html#%D0%94%D0%BB%D1%8F-%D0%BF%D1%80%D0%BE%D0%B4%D0%B2%D0%B8%D0%BD%D1%83%D1%82%D1%8B%D1%85-%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D0%B5%D0%B9)

### CLI - бойынша жұмысты үйрену

# - Тестирование ??? - бұл нәрсе не үшін керек, қалай қолданылады, төмендегі үшеуінің айырмашылығы неде

## [Введение](https://ru.vuejs.org/v2/guide/testing.html#%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5)

## Жазылған кодтарда қателер кетуі мүмкін оларды тексерудің

Жазылған кодтарда қателер кетуі мүмкін оларды тексерудің 3түрі бар

* Модульное (unit) тестирование
* Тестирование компонентов
* End-To-End (E2E) тестирование

# 

## [Модульное (unit) тестирование](https://ru.vuejs.org/v2/guide/testing.html#%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5-unit-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)

### #[Введение](https://ru.vuejs.org/v2/guide/testing.html#%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-1)

### Бірлік сынақтары кодтың жеке бөліктерін оқшаулап тексеруге мүмкіндік береді

### Басқа тексеру типтерінен айырмашылығы жоқ

### #[Выбор фреймворка](https://ru.vuejs.org/v2/guide/testing.html#%D0%92%D1%8B%D0%B1%D0%BE%D1%80-%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA%D0%B0)

### Сәтсіз сынақ туралы жай ғана хабарлаумен қатар, мұндай кітапханалар күтілетін және нақты нәтижелер арасындағы айырмашылық сияқты сынақтың неге сәтсіз болғанын түсіндіретін мәтінмән береді.

### Сондай-ақ, белсенді қауымдастықтың артықшылығы мынада, ол қиындықтарға кез келген кезде үлкен қолдау көрсете алады.

### #[Фреймворки](https://ru.vuejs.org/v2/guide/testing.html#%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA%D0%B8)

### мұнда Vue.js экожүйесінде пайдаланылатын танымал бірлік сынақ құралдарының кейбірі берілген.

Jest - қарапайымдылыққа бағытталған JavaScript тестілеу жүйесі. Оның бірегей мүмкіндіктерінің бірі қолданбалы модульдерді тексерудің балама құралы ретінде суретке түсіру сынақтарын жасау мүмкіндігі болып табылады.

Mocha - бұл ептілікке бағытталған JavaScript тестілеу жүйесі. Ол шпиондық (мысалы, «Синон») немесе бекітулер (мысалы, «Чай») сияқты жалпы функцияларды орындау үшін әртүрлі кітапханаларды таңдауға мүмкіндік береді. Mocha бағдарламасының тағы бір бірегей ерекшелігі - ол сынақтарды шолғышта да, Node.js жүйесінде де орындауға мүмкіндік береді.

## [Тестирование компонентов](https://ru.vuejs.org/v2/guide/testing.html#%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BE%D0%B2)

#[Введение](https://ru.vuejs.org/v2/guide/testing.html#%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-1)

Сондықтан, пайдаланушыларға мұны сенімді түрде жасауға мүмкіндік беретін, сонымен қатар Vue бағдарламасының нақты мүмкіндіктерін қолдайтын құрамдас тестілеу құрылымдары жасалды. мысалы, Vuex, Vue Router немесе басқа Vue плагиндерімен біріктіру.

#[Выбор фреймворка](https://ru.vuejs.org/v2/guide/testing.html#%D0%92%D1%8B%D0%B1%D0%BE%D1%80-%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA%D0%B0)

Келесі бөлімде қолданбаңыз үшін қай құрамдас тестілеу жүйесі ең жақсы екенін бағалау кезінде ескерілетін нұсқаулар берілген.

.... жоғарыдағыларға ұқсас

## [End-to-End (E2E) тестирование](https://ru.vuejs.org/v2/guide/testing.html#End-to-End-E2E-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) ???

.... жоғарыдағыларға ұқсас

# - Поддержка TypeScript ???

# - TypeScript -ті үйреніп сосын осы бөлімді үйрену керек

## [Официальные файлы объявлений в npm-пакетах](https://ru.vuejs.org/v2/guide/typescript.html#%D0%9E%D1%84%D0%B8%D1%86%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5-%D1%84%D0%B0%D0%B9%D0%BB%D1%8B-%D0%BE%D0%B1%D1%8A%D1%8F%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9-%D0%B2-npm-%D0%BF%D0%B0%D0%BA%D0%B5%D1%82%D0%B0%D1%85)

Мұның бәрі npm сайтында жарияланғандықтан, Typings сияқты сыртқы құралдарды пайдаланудың қажеті жоқ, себебі түр туралы мәлімдемелер Vue көмегімен автоматты түрде импортталады.

## [Рекомендуемая конфигурация](https://ru.vuejs.org/v2/guide/typescript.html#%D0%A0%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D1%83%D0%B5%D0%BC%D0%B0%D1%8F-%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D1%8F)

// tsconfig.json

{

"compilerOptions": {

// это соответствует поддержке браузеров у Vue

"target": "es5",

// это обеспечивает более строгий вывод для свойств данных в `this`

"strict": true,

// при использовании webpack 2+ или rollup, добавить поддержку tree-shaking:

"module": "es2015",

"moduleResolution": "node"

}

}

strict: true (или, по крайней мере, noImplicitThis: true, ) қойылуы керек , типті тексеруді қолдану үшін, әйтпесе ол әрқашан кез any (келген түр) ретінде қарастырылады.

## [Инструменты разработки](https://ru.vuejs.org/v2/guide/typescript.html#%D0%98%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D1%8B-%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8)

### #[Создание проекта](https://ru.vuejs.org/v2/guide/typescript.html#%D0%A1%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0)

[**Vue CLI 3**](https://github.com/vuejs/vue-cli) позволяет генерировать новые проекты, которые используют TypeScript. Чтобы начать:

# 1. Установите Vue CLI, если она ещё не установлена

npm install --global @vue/cli

# 2. Создайте новый проект, затем выберите опцию «Manually select features»

vue create my-project-name

### #[Поддержка редакторов](https://ru.vuejs.org/v2/guide/typescript.html#%D0%9F%D0%BE%D0%B4%D0%B4%D0%B5%D1%80%D0%B6%D0%BA%D0%B0-%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%BE%D0%B2)

### TypeScript үшін тамаша кірістірілген қолдауды қамтамасыз ететін Visual Studio кодын пайдалануды ұсынамыз.

## Бір файлдық құрамдастарды пайдалансаңыз, бір файл құрамдастары ішінде TypeScript шығысын қамтамасыз ететін тамаша Vetur кеңейтімін және т.б. мүмкіндіктерін пайдаланыңыз.

## [Использование](https://ru.vuejs.org/v2/guide/typescript.html#%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)

Vue.component или Vue.extend:

import Vue from 'vue'

const Component = Vue.extend({

// вывод типов включён

})

const Component = {

// это НЕ БУДЕТ работать,

// потому что TypeScript не может определить, что это опции компонента Vue.

}

**Пример**

<template>

  <div>

    <h1>{{ msg }}</h1>

    <h1>{{ firstName }}</h1>

    <h1>{{ lastName }}</h1>

  </div>

</template>

<script lang="ts">

import { Component, Prop, Vue } from "vue-property-decoratior";

@Component

export default class HelloWorld extends Vue {

  @Prop() private msg!: string;

  firstName = "Erik";

  lastName = "Hanchett";

  get fullName() {

    return this.firstName + this.lastName;

  }

}

</script>

## [Компоненты Vue на основе классов](https://ru.vuejs.org/v2/guide/typescript.html#%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D1%8B-Vue-%D0%BD%D0%B0-%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%B5-%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%BE%D0%B2) ???

**'vue-class-component'- ті қолданып класс ты бірнәрсе қылады**

import Vue from 'vue'

import Component from 'vue-class-component'

// декоратор @Component указывает, что класс — это компонент Vue

@Component({

// здесь можно использовать все опции компонента

template: '<button @click="onClick">Click!</button>'

})

export default class MyComponent extends Vue {

// Данные инициализации могут быть объявлены как свойства экземпляра

message: string = 'Hello!'

// Методы компонента могут быть объявлены как методы экземпляра

onClick (): void {

window.alert(this.message)

}

}

## [Расширение типов для использования с плагинами](https://ru.vuejs.org/v2/guide/typescript.html#%D0%A0%D0%B0%D1%81%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%B8%D0%B5-%D1%82%D0%B8%D0%BF%D0%BE%D0%B2-%D0%B4%D0%BB%D1%8F-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D1%81-%D0%BF%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8) ???

Плагиндер Vue жүйесіне жаңа жаһандық сипаттарды, дана сипаттарын және құрамдас параметрлерін қоса алады. Мұндай жағдайларда, TypeScript-ке компиляциялау үшін плагинді қосу үшін түр туралы мәлімдемелер қажет. (Плагиндерді Vue -ге қосуға болады,) ал TypeScript-ке қосу үшін **[module augmentation](https://www.typescriptlang.org/docs/handbook/declaration-merging.html" \l "module-augmentation" \t "_blank)** қолданылдаы

## [Аннотации возвращаемых типов](https://ru.vuejs.org/v2/guide/typescript.html#%D0%90%D0%BD%D0%BD%D0%BE%D1%82%D0%B0%D1%86%D0%B8%D0%B8-%D0%B2%D0%BE%D0%B7%D0%B2%D1%80%D0%B0%D1%89%D0%B0%D0%B5%D0%BC%D1%8B%D1%85-%D1%82%D0%B8%D0%BF%D0%BE%D0%B2) ???

TypeScript Vue типтік мәлімдемелерінің циклдік сипатына байланысты белгілі әдістердің түрлерін анықтауда қиындықтарға тап болуы мүмкін.

Осы себепті, көрсету және есептелетіндер сияқты әдістер үшін қайтару түріне түсініктеме беру қажет болуы мүмкін.

## [Аннотация типов входных параметров](https://ru.vuejs.org/v2/guide/typescript.html#%D0%90%D0%BD%D0%BD%D0%BE%D1%82%D0%B0%D1%86%D0%B8%D1%8F-%D1%82%D0%B8%D0%BF%D0%BE%D0%B2-%D0%B2%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D1%85-%D0%BF%D0%B0%D1%80%D0%B0%D0%BC%D0%B5%D1%82%D1%80%D0%BE%D0%B2) ???

# -Публикация на production

## [Включение режима production](https://ru.vuejs.org/v2/guide/deployment.html#%D0%92%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B5-%D1%80%D0%B5%D0%B6%D0%B8%D0%BC%D0%B0-production)

Vue жалпы қателер мен тұзақтарды шешуге көмектесу үшін әзірлеу процесінде көптеген пайдалы ескертулер береді. Бірақ бұл ескертулер кейде пайдасыз, тіпті зиян болуы мүмкін

### #[Без использования систем сборки](https://ru.vuejs.org/v2/guide/deployment.html#%D0%91%D0%B5%D0%B7-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC-%D1%81%D0%B1%D0%BE%D1%80%D0%BA%D0%B8) ???

### #[С использованием систем сборки](https://ru.vuejs.org/v2/guide/deployment.html#%D0%A1-%D0%B8%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%D0%BC-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC-%D1%81%D0%B1%D0%BE%D1%80%D0%BA%D0%B8) ???

### барлық тексерулер мен ескертулерді жою үшін осы айнымалы мәнді қайта анықтау мүмкіндігін береді.

#### Webpack

#### Browserify

#### Rollup

## [Прекомпиляция шаблонов](https://ru.vuejs.org/v2/guide/deployment.html#%D0%9F%D1%80%D0%B5%D0%BA%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%86%D0%B8%D1%8F-%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%BE%D0%B2) ???

Шаблонды алдын ала құрастыру

DOM немес JavaScript -те шаблон құрастырған кезде, компиляция в render-функциясы жылдам орындалады, бірақ приложение мықты болу үшін  **render**-функциясынан аулақ болған дұрыс

Шаблонды алдын ала құрастырудың ең қарапайым түрі [**однофайловые компоненты**](https://ru.vuejs.org/v2/guide/single-file-components.html).

Шаблонды алдын ала құрастыруда [**vue-template-loader**](https://github.com/ktsn/vue-template-loader) қолдануға болады

## [Извлечение CSS из компонентов](https://ru.vuejs.org/v2/guide/deployment.html#%D0%98%D0%B7%D0%B2%D0%BB%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5-CSS-%D0%B8%D0%B7-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BE%D0%B2) ???

однофайловых компонентов кезінде <style> бөлігі автоматты түрде пайда болады, баирлығында мұндай болуы уақытты алуы мүмкін, одан құтылу үшін, оладыбір файлға извлечить жасаған дұрыс, оны қалай жасаудың тәсілдері төменде көрсетілген

* [**Webpack + vue-loader**](https://vue-loader.vuejs.org/ru/configurations/extract-css.html) (в vue-cli с шаблоном webpack это уже настроено)
* [**Browserify + vueify**](https://github.com/vuejs/vueify#css-extraction)
* [**Rollup + rollup-plugin-vue**](https://vuejs.github.io/rollup-plugin-vue/#/en/2.3/?id=custom-handler)

## [Отслеживание ошибок во время выполнения](https://ru.vuejs.org/v2/guide/deployment.html#%D0%9E%D1%82%D1%81%D0%BB%D0%B5%D0%B6%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BE%D1%88%D0%B8%D0%B1%D0%BE%D0%BA-%D0%B2%D0%BE-%D0%B2%D1%80%D0%B5%D0%BC%D1%8F-%D0%B2%D1%8B%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F) ???

Орындау уақытында қателерін қадағалау

Компонентті көрсету кезінде қате орын алса, глобальныйдағы Vue.config.errorHandler функциясына жіберіледі. Бұл ілмекті Sentry сияқты қателерді бақылау қызметімен бірге пайдалануды көрсетеді,