**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

## [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)

Сенімді қосымшаларды құру туралы айтатын болсақ, тесттер жеке адамның да, команданың да қабілетінде маңызды рөл атқара алады, жаңа функционалдылық жасайды, кодты қайта өңдейді, қателерді түзетеді және т. б. тестілеудің көптеген тәсілдері болса да, веб-қосымшалар контекстінде үш категория бөлінеді:

* Модульное (unit) тестирование ( жеке бөліп тексеру)
* Тестирование компонентов(Vue қолдайтын компоненттерді сынау үшін)
* End-To-End (E2E) тестирование( барлық қабаттарын бірден тексереді)

# 

Бұл бөлімнің мақсаты-тестілеудің экосистемасын бағыттауға және Vue қосымшасы немесе компоненттер кітапханасы үшін қолайлы құралдарды таңдауға көмектесу

# - бұл нәрсе не үшін керек - кодтардағы қателер бар жоғын тексету, тексеріп отыру үшін қолданылатын библиотекалар типтері мен түрлер

## [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" \l "%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-1" \o "Введение)

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

### Басқа тексеру типтерінен айырмашылығы жоқ

### мақсаты:

### - әзірлеушілерге кодқа деген сенімділікті қамтамасыз ету

### - мұқият және мазмұнды тесттер жазу

### - жаңа мүмкіндіктерді әзірлеу немесе кодты қайта өңдеу

### осыларды орындау кезінде қосымшаның Функционалды және тұрақты болып қалатынына сенімді болуға мүмкіндік береді.

### #[Выбор фреймворка](https://ru.vuejs.org/v2/guide/testing.html" \l "%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" \o "Выбор фреймворка)

### 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" \l "%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA%D0%B8" \o "Фреймворки)

### 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.

Бір тестілеуді орындауға Мысал, басқа тестілеуді де осылай орындаға болады :

1) Npm көмегімен ғаламдық деңгейде орнатыңыз:

$ npm install --global mocha

2) жұмысқа кірісу

$ npm install mocha

$ mkdir test

$ $EDITOR test/test.js # or open with your favorite editor

3) Сіздің редакторыңызда:

var assert = require('assert');

describe('Array', function() {

describe('#indexOf()', function() {

it('should return -1 when the value is not present', function() {

assert.equal([1, 2, 3].indexOf(4), -1);

});

});

});

4) Терминалға ораламыз:

$ ./node\_modules/mocha/bin/mocha

Array

#indexOf()

✓ should return -1 when the value is not present

1 passing (9ms)

5) package.json файлында тест сценарийін орнатыңыз :

"scripts": {

"test": "mocha"

}

6) Содан кейін сынақтарды бастаңыз:

$ npm test

## [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" \l "%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-1" \o "Введение)**

Көптеген Vue құрамдастарын сынау олардың функционалдығын( жұмысын) толық тексере алу үшін оларды DOM жүйесіне (виртуалды немесе нақты) орнатуды талап етеді. Бұл тұжырымдама сонымен қатар құрылымдық агностикалық болып табылады. Сондықтан пайдаланушыларға оны сенімді түрде жасауға мүмкіндік беретін, сонымен қатар Vuex, Vue Router немесе басқа Vue плагиндерімен интеграция сияқты Vue мүмкіндіктерін қолдайтын компоненттерді сынау үшін құрамдас тестілеу құрылымдары жасалды.

**#[Выбор фреймворка](https://ru.vuejs.org/v2/guide/testing.html" \l "%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" \o "Выбор фреймворка)**

Келесі бөлімде сіздің бағдарламаңыз үшін компоненттерді сынаудың қай негізі жақсы екенін бағалау кезінде ескеру қажет нұсқаулар берілген

**Vue экосистемасымен ыңғайлы үйлесімділік**

Компоненттерді тестілеу кітапханасы Vue экожүйесімен мүмкіндігінше ыңғайлы,үйлесімді болуы керек. Бұл Vuex, vue Router және бағдарламада қолданылатын кез-келген басқа vue плагиндерін тексеруге мүмкіндік береді.

**Бірінші дәрежелі қате туралы есептер**

Сынақтар сәтсіз болғанда, компонентті тестілеу жүйесі мәселені түзетуге, кететін уақытты азайтуға көмектесетін пайдалы, қате ақпаратын қамтамасыз ету өте маңызды. Сәтсіз сынақ туралы жай ғана хабарлаумен қатар, мұндай кітапханалар күтілетін және нақты нәтижелер арасындағы айырмашылық сияқты сынақтың неге сәтсіз болғанын түсіндіретін мәтінмән береді.

### #[Рекомендации](https://ru.vuejs.org/v2/guide/testing.html" \l "%D0%A0%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D0%B8%D0%B8" \o "Рекомендации)

### Vue Testing Library

#### Vue Test Utils

## [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)

E2E тестілері сіздің бағдарламаңыздың барлық қабаттарын бірден тексереді. Бұған тек алдыңғы код қана емес, сонымен қатар пайдаланушылар жұмыс істейтін барлық байланысты қызметтер мен сервер инфрақұрылымының жұмысы кіреді. Олардың қосымшамен қалай әрекеттесетінін тексере отырып, E2E тестілері көбінесе қосымшаның дұрыс жұмыс істейтініне сенімді болудың кілті болып табылады.

### # [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%B0-2)

### Тарихи тұрғыдан алғанда, end-to-end (E2E) интернеттегі тестілеу сенімсіз (flaky) сынақтар мен даму процестерінің баяулауына байланысты теріс беделге ие болды. Қазіргі заманғы E2E құралдары сенімді, интерактивті және пайдалы сынақтарды құруда алға жылжыды. E2E тестілеу үшін жақтауды таңдау үшін келесі бөлімде қосымшаны тестілеу үшін жақтауды таңдау кезінде ескеру қажет бірнеше нұсқаулар берілген.

### Кроссбраузерлік тестілеу

### End-to-end (E2E) тестілеуінің басты артықшылықтарының бірі-қосымшаның жұмысын бірнеше шолғышта тексеру мүмкіндігі. Браузердің 100% қамтуы қажет болып көрінуі мүмкін, бірақ бұл әр браузерде оларды дәйекті түрде іске қосу үшін қажет машинаның қосымша уақыты мен қуатына байланысты команда ресурстарының қайтарымдылығын төмендететінін ескерген жөн. Нәтижесінде, қосымшаның мүмкіндіктерін тексеру үшін кроссбраузер сынақтарының санын таңдау кезінде ымыраға келу туралы ұмытпау керек.

### E2E браузерлерге қатысты мәселелерді шешудің танымал әдістерінің бірі-жиі қолданылмайтын браузерлер үшін (мысалы, < IE11, Safari-дің ескі нұсқалары және т.б.) қосымшаларды бақылау және қате туралы хабарлау құралдарын пайдалану (мысалы, Sentry, LogRocket, т. б.).

### Жылдам кері байланыс циклдары

### Тесттер мен дамудың end-to-end (E2E) негізгі мәселелерінің бірі-бүкіл тест жиынтығын іске қосу көп уақытты қажет етеді. Әдетте, олар тек pipelines үздіксіз Интеграция және орналастыру (CI/CD) іске қосылады. Қазіргі заманғы E2E рамалары параллелизация арқылы бұл мәселені шешуге көмектеседі, бұл CI/CD pipelines-ті тездетуге мүмкіндік береді. Сонымен қатар, жергілікті даму кезінде сіз қазір жұмыс істеп жатқан Бет үшін сынақтарды, сондай-ақ әзірлеушілердің өнімділігін арттыратын сынақтарды ыстық қайта жүктеу мүмкіндігін таңдай аласыз.

### Бірінші класты жөндеу тәжірибесі

### Әзірлеушілер әдетте анықтау үшін терминал терезесіндегі журналдарды бақылауға сүйенеді бұл дұрыс болмады, қазіргі заманғы end-to-end (E2E) жақтаулар өздеріне таныс құралдарды ұсынады, мысалы, шолғышты жасаушы құралдарын пайдалану.

### Бассыз (headless) режимде көріну

### Үздіксіз интеграция / орналастыру құбырларында end-to-end (E2E) тестілерін іске қосқан кезде, олар көбінесе бассыз шолғыштарда орындалады (яғни, шолғыштың көрінетін терезесі көрінбейді, яғни байқау үшін қойылатын шолғыш терезелері). Нәтижесінде, қателер пайда болған кезде, қазіргі E2E жақтаулары қолдайтын маңызды қасиет-қатенің пайда болу себебі туралы түсінік алу үшін тестілеудің әртүрлі кезеңдерінде сіздің қолданбаңыздың суреттерін және/немесе бейнелерін көруге бірінші дәрежелі қолдау. Тарихи тұрғыдан алғанда, мұндай интеграцияны сақтау өте шаршатады.

### # [Рекомендации](https://ru.vuejs.org/v2/guide/testing.html#%D0%A0%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D0%B8%D0%B8-1)

#### Cypress.io

#### Nightwatch.js

#### Puppeteer

#### TestCafe

# - 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

Бұл тақырып код жазып жатқанда ескертулер беріп отырады, берілген ескертулерді шешу жолдары

Vue CLI пайдалансаңыз, төмендегі кеңестердің көпшілігі қазірдің өзінде қамтылған (сізге керек емес ). Бұл бөлім өзіңіздің құрастыру конфигурацияларыңызды пайдалансаңыз ғана қатысты.

## [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)"

### Ең қарапайым Vue қолдану түрі

### Егер сіз толық құрастыруды пайдалансаңыз, яғни. Vue жүйесін ешбір құрастыру жүйелерін пайдаланбай тікелей <script> тегі арқылы қосыңыз, өндіріс үшін кішірейтілген нұсқаны (vue.min.js) пайдаланғаныңызға көз жеткізіңіз. Екі нұсқаны орнату бетінен табуға болады.

### # [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)"

### Ескертулері, қателерді жоюға көмектесетін жинап тексерулер , көптеген Js файлдардың барлығы бір файлға салып, бір жерден тексеру, бақылау

### Барлық vue-cli шаблондары сіз үшін қазірдің өзінде орнатылған, бірақ оның қалай жасалатынын білу пайдалы болар еді:

### Ресурстар:

#### Webpack (кодтрадың жазылу өте қиын, интернеттен көшірген дұрыс)

#### 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 (қателер шықпауы үшін оңтайландыру)

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)**.

Егер сіз Webpack-ті қолдансаңыз және JavaScript пен шаблон файлдарын бөлгіңіз келсе, vue-template-loader-ді қолдана аласыз, ол шаблон файлдарын құрастыру кезеңінде JavaScript-тегі render функцияларына түрлендіреді.

## [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, олады бір файлға шығару (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 пайдалану жақсы идея болуы мүмкін-әсіресе Vue интеграциясы ресми түрде қолдау тапқандықтан.

Vue.config.errorHandler Vue қателерді ескертіп отыру.

Vue.config.silent = true Vue журналдары мен ескертулерін өшіру.