*87029329156*

*JUNIOR STRONG*

vuex ???

router: /id:2 ???

Router ???

Vuex ???

HTML +CSS өткен тақырыптарды реттеп қою(тез қайталауға ыңғайлы болса болды)

Vue бойынша өткен тақырыптарды реттеп қою(тез қайталауға ыңғайлы болса болды)

Vue API -жұмыс, Api ???

token ??? (тез қайталауға ыңғайлы болса болды)

**JavaScript Vue - Прдвинутный уревен,** JSX, mixin

API термины , [Категории правил](https://ru.vuejs.org/v2/style-guide/index.html#%D0%9A%D0%B0%D1%82%D0%B5%D0%B3%D0%BE%D1%80%D0%B8%D0%B8-%D0%BF%D1%80%D0%B0%D0%B2%D0%B8%D0%BB)а, JS-опшый

requery ???

**MIDDLE**

Махамбет практикасынан мысалдар, библиотекалар, білгендерінің барлығы

Жұмыс істеу үшін нелер керек болады, нелер үйрену керек git lap сияқты

router, vuex - Продвинутые темы

// strict: true,???

API - Продвинутые темы

JavaScript - Средный уревен

https://www.youtube.com/watch?v=uLY9GXGMXaA  **~ ES6 PRO + class**

ES6 pro (classes, symbol, generators\_for, reflect, proxy)

https://artelogic.net/blog/what-is-the-difference-between-junior-middle-and-senior-developers-software-engineers **(**[***создание CRM системы***](https://www.youtube.com/playlist?list=PLqKQF2ojwm3njNpksFCi8o-_c-9Vva_W0)***) Практика***

**Leorn, Mdn**

[Категории правил](https://ru.vuejs.org/v2/style-guide/index.html#%D0%9A%D0%B0%D1%82%D0%B5%D0%B3%D0%BE%D1%80%D0%B8%D0%B8-%D0%BF%D1%80%D0%B0%D0%B2%D0%B8%D0%BB)а, БЭМ, html кодстайл , className, барлығын қатесіз дұрыс жазу

Оптимизация данных

**MIDDLE** *STRONG* **- УРЕВЕН АНЫҚТАУ**, нелер үйрену керек(қысқарған тәсілдер)

html : Pug , SASS, Bootstrap осылар сияқты основной тілдерден хабарлы болу

https://acmp.ru/index.asp?main=task&id\_task=108 -олимпияда

Түрлі практикалық мысалдар ????

    async onSubmit () {

      await this.$api.post('/user/login', {

        email: this.email,

        password: this.password // login, parol - енгізіп api-ге доступ алады, сосын api әр сұрағанда әрүрлі token жібереді

      })

        .then((response) => {

          Cookies.set('clagme-tk', response.data.data.token) // сосын ол token- ді Cookies-қа сақтап қоямыз

}

}

config.headers['Authorization'] = 'Bearer ' + Cookies.get('clagme-tk')

//әрбір api запроста token жіберу керек сол үшін оны глобалныйға осылай салып қояды

// барлығында ['Authorization'] = 'Bearer ' болмайды 'Authorization' - // бас0аша болуы мүмкін оны postman->headers бөлімінен көруге болады,

// егер 'Bearer болмауы керек болса оны жазбаяқ қояды

//Мысалы былай:config.headers['Authorization'] = Cookies.get('clagme-tk')

**FormData.append**

var formData = new FormData(); // Currently empty

formData.append(name, value);

formData.append(name, value, filename)

formData.append('username', 'Chris');

formData.append('userpic', myFileInput.files[0], 'chris.jpg');

 console.log(Array.from(formDatas));

      for (let key of formDatas) {

        console.log(key);

      }

[**FormData.delete()**](https://developer.mozilla.org/ru/docs/Web/API/FormData/delete)

formData.delete('username');

**FormData.get()**

formData.append('username', 'Борис');

formData.append('username', 'Кирилл');

**То метод get() вернёт только первое добавленное значение username:**

formData.get('username'); // Вернёт "Борис"

[**FormData.getAll()**](https://developer.mozilla.org/ru/docs/Web/API/FormData/getAll)

formData.append('username', 'Борис');

formData.append('username', 'Кирилл');

formData.getAll('username'); // Вернёт ["Борис", "Кирилл"]

[**FormData.has()**](https://developer.mozilla.org/ru/docs/Web/API/FormData/has)

formData.has('username'); // Returns false

formData.append('username', 'Chris');

formData.has('username'); // Returns true

[**FormData.set()**](https://developer.mozilla.org/ru/docs/Web/API/FormData/set)  
*бар кілтке жаңа мән тағайындайды немесе бұрын орнатылмаған болса, кілт/мәнді қосады.*

formData.set(name, value);

formData.set(name, value, filename);

***set ()*** *мен* ***FormData.append*** *арасындағы айырмашылық: егер берілген кілт бұрыннан бар болса*

***-set ()*** *ондағы деректерді жаңасымен ауыстырады,*

***-FormData.append*** *соңында қалғанына жаңа мән қосады.*

**formData.keys();**

var formData = new FormData();

formData.append('key1', 'value1');

formData.append('key2', 'value2');

for (var key of formData.keys()) {

console.log(key);

}

Результат: key1, key2

**formData.entries();**

var formData = new FormData();

formData.append('key1', 'value1');

formData.append('key2', 'value2');

for(var pair of formData.entries()) {

console.log(pair[0]+ ', '+ pair[1]);

}

Результат:

key1, value1

key2, value2

**formData.values();**

var formData = new FormData();

formData.append('key1', 'value1');

formData.append('key2', 'value2');

for (var value of formData.values()) {

console.log(value);

}

Результат:

value1, value2

**VUE - ТЕОРИЯ**

**beforeCreate()** : data() әлі доступный емес, DOM дерево әлі жоқ

**created()** : data() уже доступ бар , DOM дерево әлі жоқ

(created: back-end - ке запрос жіберетін болса осы дұрыс )

***html***: DOM дерево

**beforeMount()** : DOM дерево бар, бірақ одан данный алуға, оны өзгертуге доступ жоқ

**mounted()** : DOM дерево бар, одан данный алуға, оны өзгертуге доступ бар

(mounted: *innerHTML, quertySelection(#id***) -** сияқтыға доступ**)**

**beforeUpdate, updated** - когда данный изменены , обновление DOM дерево, қосымшаның өзгеруі

кнопканны басқанда немесе қандайда бір жағдайда DOM дерево (html) өзгеруі (өшіп, сосыын қайта өзгеріп барып қосылу емес, өшпей өзгеруі)

*beforeUpdate()* : сол данный өзгергенге дейін

*updated()* : сол данный өзгергеннен кейін

**beforeDestroy()** : удалит дочерный компонент, наблюдател, случатели собития ????

( компонент-тегі DOM дерево өшсе қосылады )

**destroyed():** DOM дерево өшеді, басқа строницаға ауысқанда, немесе данныйлар, кодтар өзгеріп обновление болғанда(өшіп, сосыын қайта өзгеріп барып қосылу) , немесе кнопка басылып обновление болғанда, яғни қашан DOM дерево өшіп қайта қосылғанда орындалады ????

**methods()** - көп қайталанатын функциларда (қайта-қайта қосу не өзгерту кнопкаларында, функцияларында қолданылады) DOM дерево бөлімінде функциядан кейін () *қою жақсы* қоймасада орындалады

**computed()** - бір рет қарна орындалатын функцияларда ( КЕШ - те сақтап қалатын қайта-қайта өзгерту қажет боламамайтын функцияларда ) қолданылады. DOM дерево бөлімінде функциядан кейін () *қоюмау* *жақсы* қойса орындалмайды ????

**watch(),** - бір переменныйдің өзгеруін анықтай алады,сол переменный өзгеру кезіндегі функция,

(бұл жерде бір переменныйді функция ретінде алады)

  watch: {

    success() {

      console.log("seve");

      this.success2 = this.success2 + 10;

    },

  },

filters(), - белгілі бір нақты жерлерде өзгерістер жасау үшін

    <h1>{{ success | capitalize }}</h1>

filters: {

    capitalize(value) {

      // if (!value) return "";

      value = value.toUpperCase();

      return value.charAt(0).toUpperCase() + value.slice(1);

    },

  },

бұл жерде value деген | белгісінен алдынғы данный

бір емес бірнеше filter қатар қолдануға болады ( | filterА | filterB )

**is-component** : компоненттерді қайта-қайта жаза бермеей переменный арқылы функциямен керектісін ғана шығарып отыру

**keep-alive** - деген бір компоненттен екіншісіне өткен кезде жазылған( мысалы input - та ) данныйларды сақтап қалу

https://www.youtube.com/watch?v=09n2945JW\_0

<template>

  <div>

    <keep-alive>

        <tr v-bind:is="component"></tr>

    </keep-alive>

    <button @click="ComponentCreate()">Companent Create</button>

  </div>

</template>

<script>

import BlogPostRow from "@/components/BlogPostRow";

import NewBlock from "@/components/NewBlock";

export default {

  data() {

    return {

      component: "blog-post-row",

      componen: "new-block",

    };

  },

  components: {

    BlogPostRow,

    NewBlock,

  },

  methods: {

    ComponentCreate() {

      this.component === "new-block"

        ? (this.component = "blog-post-row")

        : (this.component = "new-block");

    },

  },

};

</script>

**vue.companent**

локалный элементтер жинағы

new Vue({

  el: "#app",

  data: {

    massege: "Hello Worlad",

  },

  mathods: {

    massegeAdd() {

      this.massege = "KUANYSH";

    },

  },

});

глобальный элементтый

Vue.filter("capatilize", function (value) {

  return value.toUpperCase();

});

глобалный компонент

Vue.component("app-car", {

  data: function () {

    return {

      massege: "Hello Worlad",

    };

  },

  template: "<div><h1>{{massege}}</h1></div>",

});

new Vue({

  el: "#app",

});

Array.prototype.**map**() **????**

 arr: [

        { name: "asd1", num: 34 },

        { name: "asd90", num: 12 },

        { name: "asd32", num: 45 },

        { name: "asd23", num: 29 },

        { name: "asd12", num: 78 },

      ],

    this.array = this.arr.map((i,) => i.num);

массив ішіндегі **керектілерін** ғана алып басқа массивке орнату

this.array2 = this.array.map((\_age) => \_age - 12);

массив ішіндегі **элементтерді** жеке-жеке қолдану басқа массивке салу

 this.array = this.arr.map((i, index) => {

      i.num;

      console.log( index);

    });

массив ішіндегі **index** номерлерін анықтау

const array = [1, 2, 3, 4, 5];

function multBy(arr, n) {

  return arr.map(function (i) {

    return i \* n;

  });

}

multBy(array, 5);

.map() бойынша функция беру арқылы тиімді орындауғада болады

Array.prototype.multBy = function (n) {

  return this.map(function (i) {

    return i \* n;

  });

};

array.multBy(5);

былай жасауғада болады глобалный тиімді тәсіл

**console.log -**

**functionName()**

**window**

**window.alert("Hello world")**

**person // обект шақыруға болады**

**person.name // обект ішіндегі элемент шақыруға болады**

**bind(window) // bind деген жақша ішіндегі обектке кіріп барады**

**bind(documrnt) // DOM элемент ішіне кіріп барады**

**bind(object)**

const person = {

  name: "Kuanysh",

  age: 23,

  sayHello: hello,

  sayHelloWindow: hello.bind(window),

  loginfo: function () {

    console.log(`Name is ${this.name}`);

    console.log(`Age is ${this.age}`);

  },

};

const lena = {

  name: "lena",

  age: 22,

};

person.loginfo.bind(lena)(); // бұл жерде bind басқа обьектке кіріп барды

**console.group() :**

console.group(`${this.name} info:`);

    console.log(`Name is ${this.name}`);

    console.log(`Age is ${this.age}`);

    console.groupEnd();

person.loginfo.bind(lena, "Бірінші переменный", "екінші")();

// loginfo функциясына переменный жібермекші болса

person.loginfo.call(lena, "Бірінші переменный", "екінші");

// функцияны шақырмаяқ () қоймаяқ сразу орындау .шін call қолданады

person.loginfo.apply(lena, ["Бірінші переменный", "екінші"]);

// apply - да call сияқты бірақ оның ішінде тек екі элемент болдаы,

сол үшін екіншісі массиф болады

**Apple** ???

**bind, call , apply - бір біріне ұқсас функциялар**

https://www.youtube.com/watch?v=UGapN-hrekw - JS console. map, buld, call ,apply

Event Loop - SetTimout -та 0 сикунт болсада синхронный кезде бәрінен кейін болады, өйткені ол Event Loop-қа кетіп шамалы кідіреді,

button-да сондай шамалы кідіруі Event Loop -та болады, нешерет басса соншарет Event Loop -та қайталаны пшақырады

npm cache verify

npm install --cache /tmp/empty-cache

with --force

A copmplete log of this run can be found in:

C:/Users\1\AppData\Romaming\npm-cache\\_logs\2021-09-08T05\_29\_29\_146Z-debug.log

npm ERR! A complete log of this run can be found in:

npm ERR! C:\Users\1\AppData\Roaming\npm-cache\\_logs\2021-09-08T05\_47\_33\_642Z-debug.

**npm install** - **node\_modules** скачать етіледі сосын ғана ***npm run serve*** болады

You are using Node v13.14.0.

Node.js 13.x has already reached end-of-life and will not be supported in future major releasses.

It's stongly recommended to use an active LTS version instead

A leter version of Node.js already installed. Setup will now exit

(егер қате болса **node.js 14** версиясы, **windows 10** орнату)

npm install --cache

npm update -g

npm install -g @vue/cli

https://nodejs.org/en/download/releases/ -Node.js Downdload

https://css-tricks.com/the-shapes-of-css/ -қисық фигуралар

https://www.flaticon.com/search?word=X&license=all&style=all&order\_by=4&type=icon

https://developer.mozilla.org/ru/docs/Web/HTML/Element/picture ???????

<picture>

<source srcset="mdn-logo-wide.png" media="(min-width: 600px)">

<img src="mdn-logo-narrow.png" alt="MDN">

</picture>

<https://codepen.io/Ba5nanas/pen/yLeJXGG> - [**Skeleton state loader**](https://codepen.io/Ba5nanas/pen/yLeJXGG)

# 

# қисық сызық , қисық фигуралар

 <div class="blocks">

        <svg class="hidden lg:block absolute right-0 inset-y-0 h-full w-48 text-white transform translate-x-1/2" fill="currentColor" viewBox="0 0 100 100" preserveAspectRatio="none" aria-hidden="true">

          <polygon points="0,0 100,0 5,100 0,100"></polygon>

      </svg>

    </div>

router - CLI да сразу таңдап орнатқан жақсы

hight - блоктара биіктік берсе көбнесе қате шығады

margin - блоктар ішінде margin -ға қатты мән беру керек

main.css -бірдей стилдерді осында жазады

flex-wrap - flex блок симаса астына көшеді

:scr ="pos.img" - моссивтегі url - енгізу

img - интернеттен ссылка қоюға болады

./ бір нүкте болса бір файл сыртқа шығады

../../file үш файл сыртқа шығады

grid-rows - қан екені белгісіз болса , көп болса жазылмайды

.conteiner - main.css жазып керек жеоінде ғана шығару

flex -көбейтіп жібермеу керек

<b> орнына <string> қолдану

Спейсимволы HTML

<div/> ішінен <div/> салу width белгілу slaider time істеу

function(){ this.swiper.slaiderNext(3000, false)} 3 секинтта ауысу

css <teg/> болйынша емес класс бойынша жазу керек тек main.css те жазу

focus жазу үшін блок алдында болу керек

App.vue файлды router -де Home -ге беруге болмайды

App.vue : router -> <router-view>

input type = file болғанда резултатты @change - мен алынады

function(e){ file = e.targe.filep[0]}

arrForEach()

Git master- Reposition "Repo"

commit: -> "Sare" - > hesh

Branch

Pull request

Merge

жылжымалы кнопка - **жасаған жаңа блоктар суреттерін жинау**

checkbox - ты span, label арқылы істеу

суретті басқанда полный экран ашу

grit - бірбіріне өтіп тұратын блоктар, жанындағы блокта көп текст болса блок созылып кетеді

............... - осындай сызықтар мазмұны сияқты істеу,

картаны қалай қолдану

Git-тегі аккаунт token -дегі аккаунт сияқты құлттанып қалады

components - жеке файлдарға бөліп қоюға болады, ат қойғанда тек таба алатын аттар қою, бөлуге болатын блоктардың барлығын бөлу керек

component - тер ішіне :bind, props[] арқылы элементтер жіберу

Бір жерде өолданатын компонент файлдарды бір паапкаға салып қою

views - атын қойғанда іздегенде табылатын

router - жеке элемент страницаларына өткенде paht: "shop/:id" қою керек "shop/2","shop/5"

сияқты кез-келгенін қабылдай береді

Верстка :

стилдер дизайнмен бірдей болу керек , тек амал жоқ кезде ғана өзгертуге болады

slider - блоктар ды анықтай алу

әр блок ұзындағын белгілеу керек

border-radius: 50% 50%

letter-spacing

transition 2s - ты hover -ге емес блок сыртына жазу керек

background-color 0.2s ease-in, color, 0,2s ease in

overflow:hidden - артығы сыртқа шығып кетпеу

word-break: break-word - ұзын сөзді бөліп астына жазу (flex-wrap сияқты)

iconka - img арқылы қойған дұрыс

position fixed - полный экранға шығару

ссылкаларды - <a/> <router/>

Шрифтерді - интернеттен тартып import қылу керек

CSS - menu -ді ul -мен істеу керек

list-style-type:none

left:50%;

transform:translate(-50%, -50%)

border-left: 10px solid #000

background:linear-gradient(#e0eDe0, #116788)

text-decoration:none <a>

text-decoration:revert <a>

list-style-image:url(img/dow-img.png)

transform: rotate(360deg)

transform:scale(15px)

transform:translateY(-10px)

transform: translateX(10px)

overflow :hidden

box-sizing:border-box

background - attachment: fixed

background-color: rgba(51,151,118,0.5)

text-decoration:line-through;

outline:none

margin: 1% auto 0px

list-style:none

flex-wrap:wrap

toggle - class name (өзгеретін)

артық div жасамау <a> ссылканың сыртынан

ценаны жазғанда класстың аты *strong* емес *price* болуы керек

letter-spacing - әріптердің арақашықтығы

col,row -деп класс қоймау керек егер 2 ғана блок болса

            onclick="btnclick(4)"

        function btnclick(index) {

          if (a == 1) {

            document.getElementsByClassName("faq\_\_text-toggle")[

              index

            ].style.display = "none";

            a = 0;

          } else {

            document.getElementsByClassName("faq\_\_text-toggle")[

              index

            ].style.display = "inline";

            a = 1;

          }

        }

markdown

oblaco

https://www.airbnb.ru/

margin, padding-сияқтылар істемесе display:block жазу керек

i5 3317u

**<video>**

**<source** src="URL"**>**

**</video>**

**<video** width="400" height="320" controls="controls" autoplay="autoplay"**>**

**<source** src="video/snowman.ogv" type='video/ogg; codecs="theora, vorbis"'**>**

**</video>**  
**<video** width="500" height="400" controls autoplay loop**>**

**<source** src="video/women.ogv" type='video/ogg; codecs="theora, vorbis"'**>**

**</video>**

http://htmlbook.ru/html/video/preload - **video** html

Npm - **Node.js** тарту керек ,

**npm** таңдау керек,

файл ішіне **npm install** арқылы **node\_module** файлын орнату

қате болып жатса өшіріп басынан орнатып шығу

**vue create vue-file**

(егер қате болса **node.js 14** версиясы, **windows 10** орнату)

**manually select features**

пробел и стрелки **таңдау**, enter

**3x** y enter

sass enter

**In package.json** enter

(save ....) n enter

**npm** enter

***Cmp -***

Win + R , cmp жазу, Enter // - команда строканы ашу

color a //изменит цвет шрифта

cls // очистит все для этого

md fileName //создать файл

copy con documentName // создать документ

copy con documentName.txt

hello world!!!!

^z

enter //ішіне тект жазып болғансоң энтр басса болды сақталады

ren document.html doc.html // изменит имя файла

ren doc.html style .css // изменит имя и тип файла

move style.css A1 // style.css файлын А1 папкасына ауыстыру

tree /f файл структурасын көру тексеру

REN C:\Users\desktop\ faile1 faile2 //папканың атын өзгерту путь арқылы

REN C:\Users\desktop\ stylr.css index.html //файл атын өзгерту путь арқылы

DIR

DIR style.css

rmdir fileName // папка удалить ету

rmdir fileName /s file2 // папка ішіндегі файл удалить ету

ver // верси авия система

systeminfo // конфигурация система

driverquery // информаци все на компьютер

**token**

авторизатсия-ның иин-і сияқты , сайтта сенің парольдарың сақтап қою қауіпті, сол үшін оның орнына **токен** сақталады

-бір сайтты регистрация істегенде дұрыс аяқталса, бәрі сәтті аяқталды деген смс пен бірге **токен** жібереді (back-end - жұмысы)

video

-сол келген **токенді** алып сайттың **cooki** -іне сақтап қою керек ??? (frond-end - жұмысы)

-кейін қайта **авторизатсия** жасағанда, анық сен екеніңді білу үшін cooki -дегі данныйды алып, input-та енгізген данныймен **тексереді**, дұрыс болса доступ, болмаса қате береді ??? (frond-end - жұмысы)

**html : Pug ,**

html

head

title

meta

body

header

div

input

footer

**SASS, базовый знание**

.header

.container

.block

.footer

.box

бір- біріне ұқсас қысқартып жазу үшін қолданады

- - ) 37 қате, БЭМ

a ссылка бетті ауыстырса,экранның ауысып өзгеруі

button отпровит ету не бірнәрсенің жағдайын өзгерту

li:not(:last-child){

margin-left: 24px;

}

https://codepen.io/meldm/pen/bqxxQV

https://www.csscodelab.com/vue-slider-carousel-code-snippet/

**сразу одаптивкаға дайын қылып окуратно істеу**

**swiper-HTML**

https://swiperjs.com/get-started

svg - icon суреттің цветін ауыстыру үшін оны img src емес HTML түрінде қойып

css- арқылы :

svg path{

fill:#fff;

}

өзгертеді

:class="{

'chats-tabs\_\_link\_active':$route.params.type==='requests',}"

егер роутер'requests' осылай болса , 'chats-tabs\_\_link\_acti осы класстар пайда болады деген сөз

Алгоритм component көп қаталанбайтын болса қолданылатын view file ішіне ғана жазуға болады

router арқылы component тер шақырғанда ең тиісді жолдың ең тиімділерін табу керек

view file ішінен component шақыру

components: {

    HeaderBlock,

    TabsBlock: () => ({

      component: import("./components/tabs.vue"),

    }),

    Messages: () => ({

      component: import("./components/messages.vue"),

    }),

  },

Alt арқылы бәрін бірге жазу shift + 1 барлығының аяғына кетеді

Alt арқылы бәрін бірге жазу home басына кетеді

Библотека ссылкаға жалғаса зогруска көп болса скачать етіп файлмен жалғау керек

svg сурет қойғанда қате шығып жатса id әр түрлі қылу керек

location.reload() - страница перезагруска жасау

**get: set:** // қою // aлу ???

функцияның ***computed:*** - ішінде болғаны үшін сырттан келген данный арқылы өзгерту қиын, оны өзгерту үшін set(newValue){} қолданылады, newValue - сырттан келген данный

  {{ fullName }}

    <input type="text" v-model="fullName" />

computed: {

    //  set(age);     // геттер:  қою     // сеттер:  aлу

    fullName: {

      get() {

        return this.firstName + " " + this.lastName + " " + this.a;

        //fullName - функциясының нәтижесінде не шығуы керек осында жазу

      },

      set(newValue) {

        // fullName - функциясының computed: - ішінде болғаны үшін сырттан келген данный арқылы өзгерту қиын, оны өзгерту үшін set(newValue){} қолданылады, newValue - сырттан келген данный

        // console.log(newValue);

        var names = newValue.split(" ");

        console.log(names);

        this.a = names[names.length - 1];

        // this.firstName = names[0];

        // this.lastName = names[names.length - 1];

      },

    },

  },

 <button class="actions\_\_reject" :style="`background:${value[0]}`">Отклонить</button>

 https://unsplash.com/ - сурет скачат

https://vcalendar.io/installation.html - календар vue-де calendar орнату

beforeEnter:function - бұл роутерде қолдануы, vuex-те, component-те басқаша қолданады

  {

    path: "/resume",

    name: "Resume",

    component: () =>import "../views/resume/resume",

    beforeEnter: isUser,

  },

beforeEnter: hasToken

afterEnter: functionName // роутер странитцасына кірерден бұрын

beforeEnter: functionName // роутер странитцасына кіргенен кейін

beforeLive... : functionName // роутер странитцасынан шығардан бұрын

const isUser = (to, from, next) => {

  let auth = Cookies.get('clagme-type')

  if(auth) next()

  else if(to.path !== '/auth/sign-in') next('/auth/sign-in')

}

to - бармақшы болған, to.path -бармақшы болған страница

from - келгені туралы ақпарат , from.path - қай страницадан келді соны көрсетеді

next('exampleLink') - жібермекші болған страница, қай страницаға лақтырғың келеді, егер

next() - жақша пустой болса келесі страница, яғни бармақшы болған страницаға (to.path) лақтырады

Cookies.get() || Cookies.set() || Cookies.remove()

import Cookies from "js-cookie";

    <button @click="example">Click</button>

    <button @click="deletes">Delete</button>

  example() {

      console.log("example");

      Cookies.set("example", "example-cookes2");

      console.log(Cookies.get("example"));

    },

    deletes() {

      Cookies.remove("example");

    },

  <picture>

      <source

        srcset="https://ichef.bbci.co.uk/news/999/cpsprodpb/15951/production/\_117310488\_16.jpg"

        media="(min-width: 600px)"

      />

      <img

        src="https://images.news18.com/ibnlive/uploads/2021/06/1622715559\_disha.jpg"

        alt="MDN"

      />

    </picture>

600px -ден төмен болса img-дегі сурет көрінеді, ал одан үлкен болса source- дегі сурет көрінеді