

Асинхронні компоненти.

Suspense

<https://ua.vuejs.org/guide/components/async.html>

<https://ua.vuejs.org/guide/built-ins/suspense.html>

Асинхронні компоненти

- завантажуються тільки за потреби у їх використанні
- дозволяють розбивати додаток на окремі чистини, які завантажуються окремо
- дозволяють прискорити завантаження додатку (завантажуються тільки ті елементи, які потрібні при старті)
- описуються з використанням *defineAsyncComponent*

Асинхронні компоненти

- завантажуються тільки за потреби у їх використанні
- дозволяють розбивати додаток на окремі чистини, які завантажуються окремо
- дозволяють прискорити завантаження додатку (завантажуються тільки ті елементи, які потрібні при старті)
- описуються з використанням *defineAsyncComponent*

Можливі шляхи отримання асинхронних компонентів

- Асинхронні компоненти на основі динамічного імпорту модуля ES 6

```
import { defineAsyncComponent } from 'vue'

const AsyncComp = defineAsyncComponent(() =>
  import('... шлях до компонента ...')
)
```

Асинхронні компоненти

- завантажуються тільки за потреби у їх використанні
- дозволяють розбивати додаток на окремі чистини, які завантажуються окремо
- дозволяють прискорити завантаження додатку (завантажуються тільки ті елементи, які потрібні при старті)
- описуються з використанням *defineAsyncComponent*

Можливі шляхи отримання асинхронних компонентів

- Асинхронні компоненти на основі динамічного імпорту модуля ES 6

```
import { defineAsyncComponent } from 'vue'

const AsyncComp = defineAsyncComponent(() =>
  import('... шлях до компонента ...')
)
```

- Динамічне створення/завантаження компонента (використовуємо *Promise*)

```
import { defineAsyncComponent } from 'vue'

const AsyncComp = defineAsyncComponent(() => {
  return new Promise((resolve, reject) => {
    // ...завантаження компонента з сервера
    resolve(/* завантажений компонент */)
  })
})
// ... використовуйте `AsyncComp`, як звичайний компонент
```

Приклад. Розробити додаток з кредитними програмами та калькулятором

Головна | Кредитні програми | Контакти

Ми допоможемо Вам вирішити ваші задачі

Ваша кредитна програма

Завантажені модулі

 LoansPage.vue?

Головна | Кредитні програми | Контакти

Ми допоможемо Вам вирішити ваші задачі

Ваша кредитна програма 1

2

 LoansPage.vue?

Головна | Кредитні програми | Контакти

Ми допоможемо Вам вирішити ваші задачі

Ваша кредитна програма

Calculator. Кредитна програма: Легка

Потрібна сума:

Загальна сума кредиту: 110

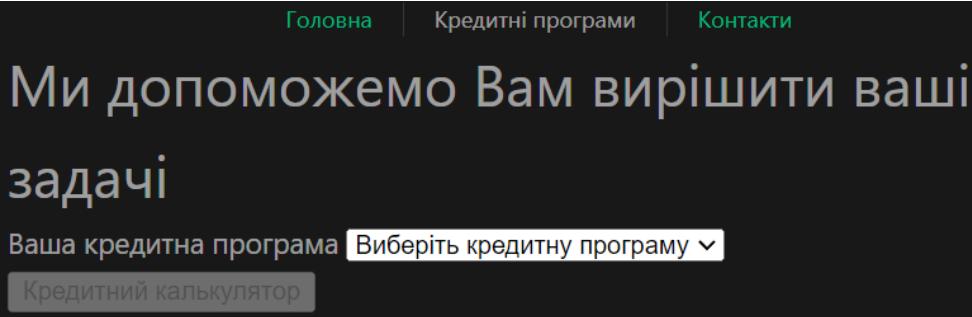
Переплата: 10

 LoansPage.vue?

 LoanCalculator.vue

Завантажується тільки за потреби

Приклад. Розробити додаток з кредитними програмами та калькулятором



```
import { defineAsyncComponent } from 'vue'

const AsyncComp = defineAsyncComponent(() =>
  import('... шлях до компонента ...')
)
```

```
<template>
  <div>
    <h1>Ми допоможемо Вам вирішити ваші задачі</h1>
    <div> Ваша кредитна програма
      <select v-model="selectedLoanProgram">
        <option value="">Виберіть кредитну програму</option>
        <option v-for="program in loanPrograms"
          :key="program.id"
          :value="program">
          {{ program.title }}
        </option>
      </select>
    </div>
    <div>
      <button
        v-if="!isCalcVisible"
        :disabled="!selectedLoanProgram"
        @click="isCalcVisible = !isCalcVisible"
      >
        Кредитний калькулятор
      </button>
      <async-loan-calculator v-else
        :credit-program-id="selectedLoanProgram.id" />
    </div>
  </div>
</template>
<script setup>
import { ref, defineAsyncComponent } from 'vue'
const AsyncLoanCalculator = defineAsyncComponent(
  () => import('@/components/LoanCalculator.vue')
)
```

Приклад. Розробити додаток з кредитними програмами та калькулятором

Головна | Кредитні програми | Контакти

Ми допоможемо Вам вирішити Ваші задачі

Ваша кредитна програма:

Calculator. Кредитна програма: Легка

Потрібна сума:

Загальна сума кредиту: 110

Переплата: 10

```
<template>
  <div>
    <h2>Calculator. Кредитна програма: {{ loanProgram.title }}</h2>
    <div>
      <label>Потрібна сума:<br/></label>
      <input v-model="loanAmount" type="number" id="loanAmount" />
    </div>
    <div>Загальна сума кредиту: {{ totalRepayment }}</div>
    <div>Переплата: {{ interestAmount }}</div>
  </div>
</template>
<script setup>
import { ref, computed } from 'vue'
import { useLoanProgramsStore } from '@/stores/loanPrograms.js'
const props = defineProps({
  creditProgramId: {
    type: Object,
    required: true
  }
})
const { getProgramById } = useLoanProgramsStore()
const loanProgram = computed(() => getProgramById(props.creditProgramId))
const loanAmount = ref(0)
const totalRepayment = computed(() => {
  return loanAmount.value + (loanAmount.value * loanProgram.value.percentage) / 100
})
const interestAmount = computed(() => {
  return totalRepayment.value - loanAmount.value
})
</script>
```

Асинхронні компоненти

Можливі шляхи отримання асинхронних компонентів

- Динамічне створення/завантаження компонента (використовуємо **Promise**)

```
import { defineAsyncComponent } from 'vue'

const AsyncComp = defineAsyncComponent(() => {
  return new Promise((resolve, reject) => {
    // ...завантаження компонента з сервера
    resolve(/* завантажений компонент */)
  })
})
// ... використовуйте `AsyncComp`, як звичайний компонент
```

Асинхронні компоненти

Можливі шляхи отримання асинхронних компонентів

- Динамічне створення/завантаження компонента (використовуємо *Promise*)

```
import { defineAsyncComponent } from 'vue'

const AsyncComp = defineAsyncComponent((() => {
  return new Promise((resolve, reject) => {
    // ...завантаження компонента з сервера
    resolve(/* завантажений компонент */)
  })
})
```

Динамічно створюємо компонент з використанням завантажених даних

```
<template>
  <h1>Найбільш популярна порода сьогодні</h1>
  <RandomDog />
</template>
<script setup>
  import { defineAsyncComponent, defineComponent } from 'vue'
  const RandomDog = defineAsyncComponent((() => {
    return new Promise((resolve, reject) => {
      fetch('https://dog.ceo/api/breeds/image/random')
        .then((response) => response.json())
        .then(resData) => {
          resolve(
            defineComponent({
              data() {
                return { resData }
              },
              template: `
                <div>
                  <h2>{{resData.message.split('/')[4]}}</h2>
                  
                </div>
              `,
            })
          )
        .catch((err) => {
          reject(err)
        })
      }
    })
  )>
</script>
```

Стани завантаження та помилки

```
const AsyncComp = defineAsyncComponent({  
  // функція-завантажувач  
  loader: () => import('./Foo.vue'),  
  
  // Компонент для використання під час завантаження асинхронного компонента  
  loadingComponent: LoadingComponent,  
  // Затримка перед показом компонента завантаження. За замовчуванням: 200 мс.  
  delay: 200,  
  
  // Компонент для використання, якщо завантаження не вдається  
  errorComponent: ErrorComponent,  
  // Компонент помилки відображатиметься, якщо встановлено  
  // та перевищено час очікування. За замовчуванням: нескінченність.  
  timeout: 3000  
})
```

Стани завантаження та помилки

```
const AsyncComp = defineAsyncComponent({  
  // функція-завантажувач  
  loader: () => import('./Foo.vue'),  
  
  // Компонент для використання під час завантаження асинхронного компонента  
  loadingComponent: LoadingComponent,  
  // Затримка перед показом компонента завантаження. За замовчуванням:  
  delay: 200,  
  
  // Компонент для використання, якщо завантаження не вдається  
  errorComponent: ErrorComponent,  
  // Компонент помилки відображатиметься, якщо встановлено  
  // та перевищено час очікування. За замовчуванням: нескінченність.  
  timeout: 3000  
})
```

Найбільш популярні породи сьогодні

Loading...



Найбільш популярні породи сьогодні

weimaraner



```
<template>  
  <h1>Найбільш популярні породи сьогодні</h1>  
  <RandomDog />  
</template>  
<script setup>  
import { defineAsyncComponent, defineComponent } from 'vue'  
import LoadingComp from './components>LoadingComp.vue'  
import ErrorComp from './components>ErrorComp.vue'  
const RandomDog = defineAsyncComponent({  
  loader: () =>  
    new Promise((resolve, reject) => {  
      fetch('https://dog.ceo/api/breeds/image/random')  
        .then((response) => response.json())  
        .then((resData) => {  
          resolve(  
            defineComponent({  
              data() {  
                return { resData }  
              },  
              template: `...`  
            })  
          )  
        })  
        .catch((err) => { reject(err) })  
    }),  
  delay: 200,  
  loadingComponent: LoadingComp,  
  errorComponent: ErrorComp,  
  timeout: 3000  
})
```

Найбільш популярні породи сьогодні

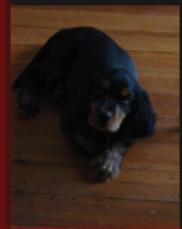
Loading...

Loading...



Найбільш популярні породи сьогодні

setter-gordon



Current Currency Rates

Currency Rate

USD 1

AED 3.67

AFN 70.46

ALL 94.36

AMD 404.92

ANG 1.79

```
<template>
  <h1>Найбільш популярні породи сьогодні</h1>
  <RandomDog />
  <CurrentCurrencyRates />
</template>
<script setup>
import { defineAsyncComponent, defineComponent } from 'vue'
import LoadingComp from './components>LoadingComp.vue'
import ErrorComp from './components>ErrorComp.vue'
const RandomDog = defineAsyncComponent({
  loader: () =>
    new Promise((resolve, reject) => {
      ...
    }),
  delay: 200,
  loadingComponent: LoadingComp,
  errorComponent: ErrorComp,
  timeout: 3000
})
const CurrentCurrencyRates = defineAsyncComponent({
  loader: () =>
    new Promise((resolve, reject) => {
      ...
    }),
  delay: 200,
  loadingComponent: LoadingComp
})
</script>
```

Suspense

`<Suspense>` — це вбудований компонент для організації асинхронних залежностей у дереві компонентів. Він може візуалізувати стан завантаження, очікуючи вирішення кількох вкладених асинхронних залежностей у дереві компонентів.

```
<Suspense>

  <асинхрний-компонент />

  <template #fallback>
    <компонент-відображення-завантаження/>
  </template>
</Suspense>
```

Suspense

Слот приймає один кореневий елемент

```
<Suspense>
  <асинхронний-компонент />
  <template #fallback>
    <компонент-відображення-завантаження/>
  </template>
</Suspense>
```

```
<template>
  <div>
    <h1>Test</h1>
    <suspense>
      <div>
        <RandomDog />
        <CurrentCurrencyRates />
      </div>
      <template #fallback>
        <loading-comp />
      </template>
    </suspense>
  </div>
</template>

<script setup>
import { defineAsyncComponent, defineComponent } from 'vue'
import LoadingComp from './components>LoadingComp.vue'

const RandomDog = defineAsyncComponent(() => {
  return new Promise((resolve, reject) => {
    ...
  })
})

const CurrentCurrencyRates = defineAsyncComponent(() => {
  return new Promise((resolve, reject) => {
    ...
  })
})
</script>
```

