??? ????? Promise?

Promise — это объект в JavaScript, который представляет **результат асинхронной операции**
(то есть того, что завершится **позже**).

Проще:

> **Promise = обещание, что значение будет, но не сразу**

????? ????? Promise

В JS много асинхронных вещей:

- запросы к серверу
- таймеры
- чтение файлов
- динамические импорты

Promise позволяет:

- дождаться результата
- обработать успех или ошибку
- не блокировать основной поток

??? ???????? Promise

```
const promise = new Promise((resolve, reject) => {
  // асинхронная работа
})
```

У Promise есть **3 состояния**:

Состояние	Что значит
`pending`	в процессе

Состояние	Что значит
`fulfilled`	успешно выполнен
`rejected`	ошибка

??????? ??????

```
const promise = new Promise((resolve) => {
  setTimeout(() => {
    resolve('Готово!')
  }, 1000)
})
```

Через 1 секунду Promise станет `fulfilled`.

??? ???????????? Promise

1?? `.then() / .catch()`

```
promise
  .then(result => {
    console.log(result)
  })
  .catch(error => {
    console.error(error)
  })
```

2?? `async / await` (??????????)

```
async function run() {
  const result = await promise
  console.log(result)
}
```

□□ `await`:

- «ставит код на паузу»
- но **не блокирует** браузер

???????? ?????? (?? ??? ?????)

```
requestClient.get('/menu/all')
```

Возвращает:

```
Promise<Menu[]>
```

Использование:

```
const menu = await getAllMenusApi()
```

??? ??????????? ??? ???????

```
console.log('1')

fetch('/api')
  .then(() => console.log('2'))

console.log('3')
```

Вывод:

```
1
3
2
```

Promise выполняется **позже**, когда стек освободится.

Promise vs ??????????? ???

Синхронный	Асинхронный (Promise)

Блокирует выполнение	Не блокирует
Сразу возвращает значение	Возвращает Promise
Просто	Гибко и эффективно

?????? ?????? ?

? ?????? `await`

```
const data = getAllMenusApi() // □ Promise
```

? ????? ???:

```
const data = await getAllMenusApi() // □ данные
```

???????

> **Promise** — это механизм работы с асинхронным кодом в JavaScript: результат будет **позже**, и JS знает, **когда и как его обработать**.

Revision #1
Created 2026-01-19 13:37:53 UTC by Nikolay
Updated 2026-01-19 13:38:50 UTC by Nikolay