

# hash vs history

## 1?? Hash mode (`#/path`)

Пример URL:

```
http://site.com/#/modules
```

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

- Всё, что после `#`, **не отправляется на сервер**
- Навигацией полностью управляет **Vue Router**
- Серверу вообще не нужно знать о роутинге

????? ?

- ✓ Работает **без настройки сервера**
- ✓ Можно открыть страницу напрямую
- ✓ Подходит для GitHub Pages, static hosting

?????? ?

- ✗ URL выглядит «грязно» (`#`)
- ✗ Хуже для SEO

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

```
createWebHashHistory()
```

## 2?? History mode (`/path`)

Пример URL:

http://site.com/modules

??? ????????

- URL **реальный**
- При обновлении браузер идёт на сервер
- Сервер должен вернуть `index.html`

????? ?

- ✓ Красивые URL
- ✓ Лучше для SEO
- ✓ Стандарт для продакшена

?????? ?

- ✗ Нужна настройка сервера
- ✗ Без неё будет 404 при обновлении

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

createWebHistory()

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

	Hash	History
URL	<code>#/modules</code>	<code>/modules</code>
Сервер	Не нужен	Нужен
SEO	Плохо	Хорошо
Обновление страницы	Всегда ок	404 без конфига
Продакшен	Редко	<b>Чаще всего</b>

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

## ? Hash — ????:

- нет доступа к серверу
- хостинг статики
- быстрый прототип
- админка без SEO

## ? History — ????:

- продакшен
- SEO важно
- есть сервер (Nginx, Apache, Node)

## ?????? ?????????????? ?????????? (Nginx)

```
location / {  
    try_files $uri $uri/ /index.html;  
}
```

## ? Nuxt

- по умолчанию **history**
- hash почти не используется

## ????????

“ **Hash** — просто и без сервера  
**History** — красиво и правильно, но требует настройки

Revision #1

Created 2026-01-17 12:22:48 UTC by Nikolay

Updated 2026-01-17 12:23:34 UTC by Nikolay