

# ПОЛНЫЙ ГИД ПО URL-РЕДИЕКТАМ

## Настройка, SEO и Виды

Визуальный справочник для  
вебмастеров и SEO-специалистов



От архитектуры трафика  
до кода в .htaccess

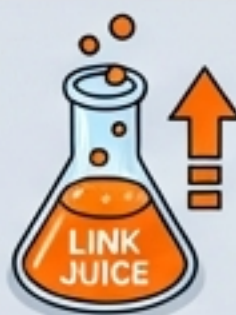
# ЧТО ТАКОЕ РЕДИРЕКТ?



**Редирект** — это серверная инструкция, автоматически отправляющая пользователя и поискового робота с запрашиваемого URL на актуальный адрес.



**UX (User Experience):** Исключение ошибок 404. Пользователь всегда находит контент.



**SEO (Link Juice):** Сохранение 90–99% ссылочного веса при правильной настройке.

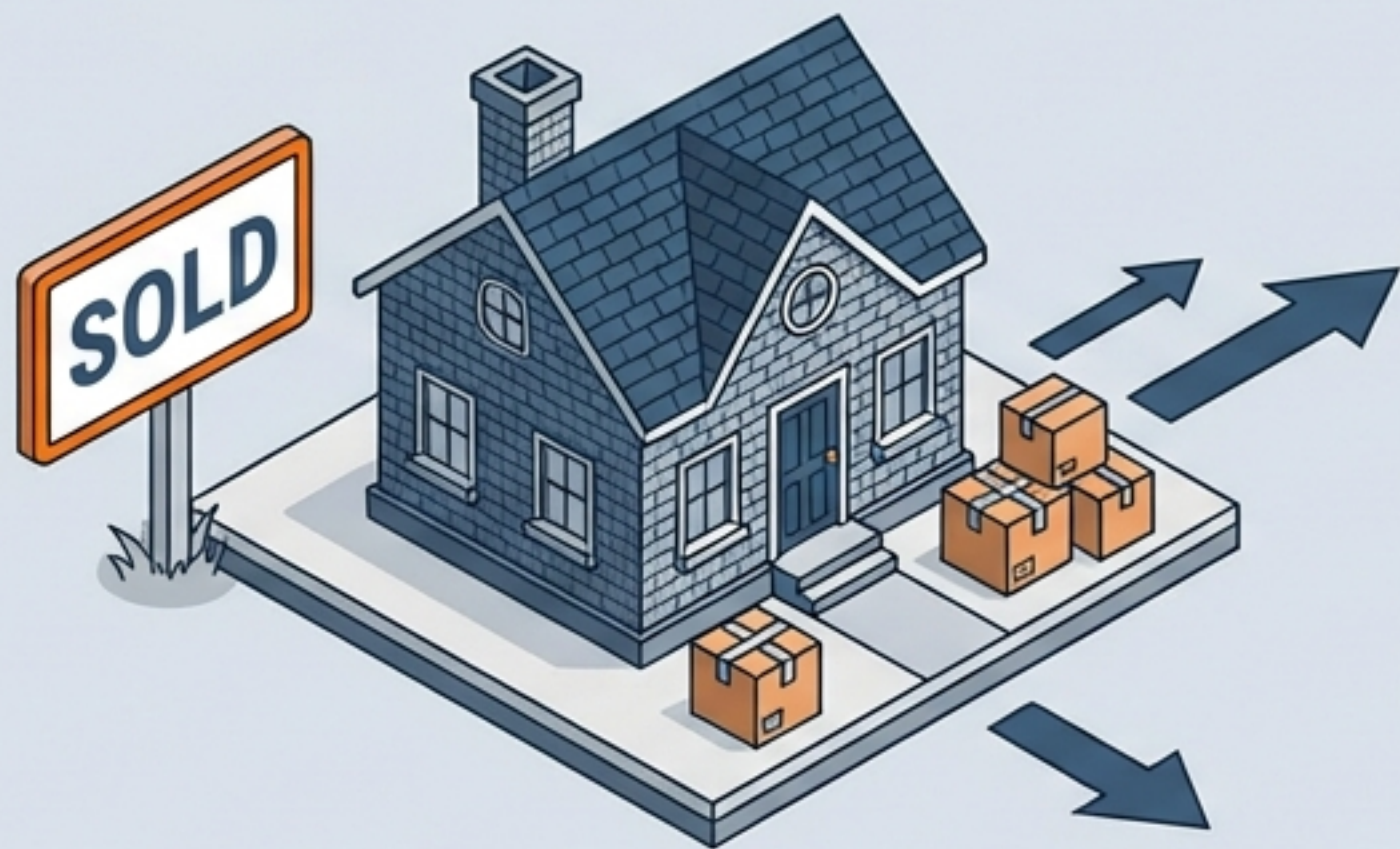


**Актуальность:** Удаление старого контента без потери трафика.

**ИНСАЙТ:** Правильный 301 редирект передает почти 100% авторитета страницы (PageRank).

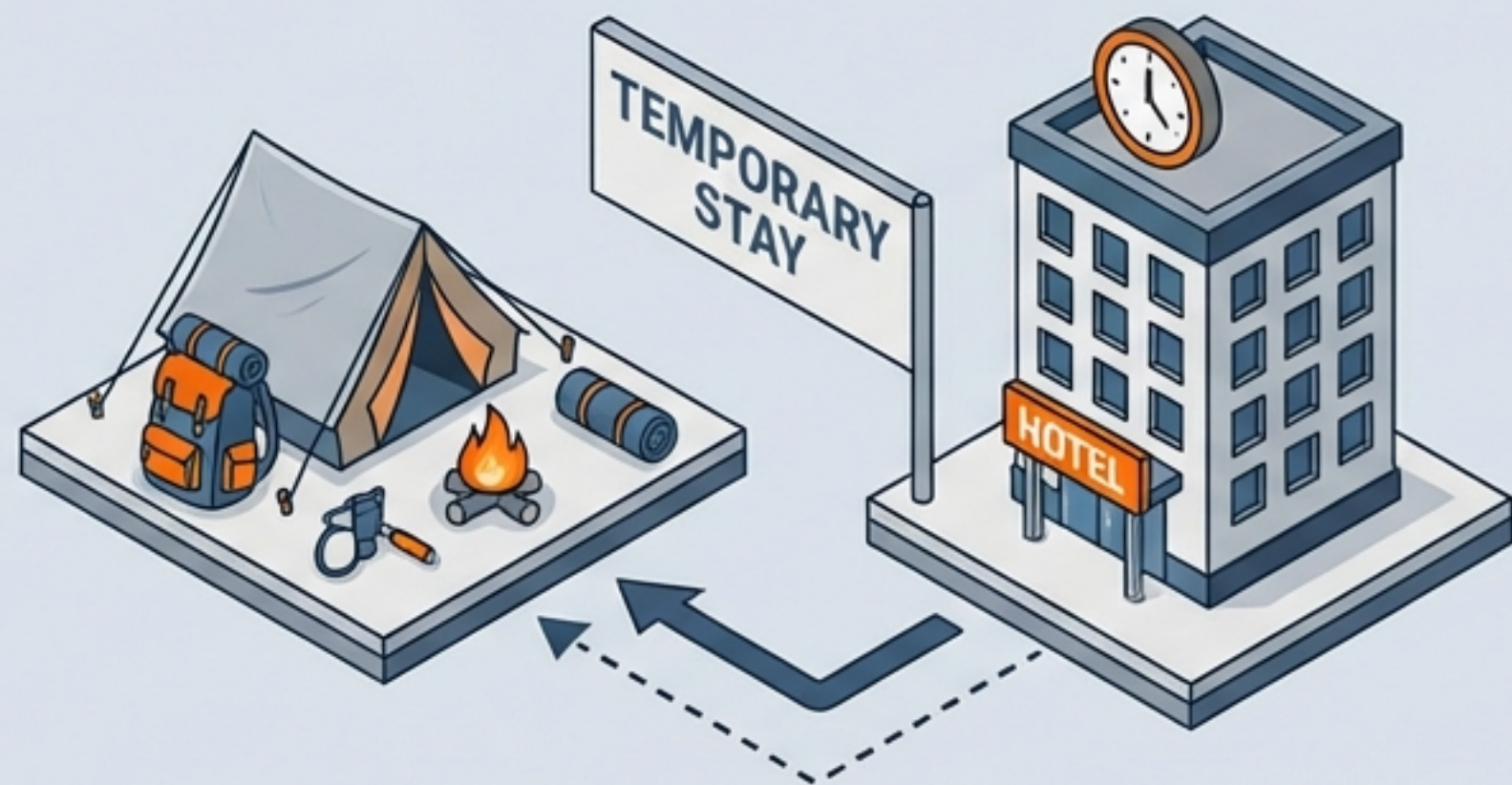
# БИТВА ТИТАНОВ: 301 VS 302

## 301: MOVED PERMANENTLY



- **Метафора:** Переезд в новый дом навсегда.
- **Действие:** Удаляет старый URL из индекса Google.
- **SEO:** Передает ссылочный вес (Link Juice).
- **Кэш:** Жесткое кэширование браузером.

## 302: FOUND / TEMPORARY



- **Метафора:** Временная ночевка в отеле.
- **Действие:** Сохраняет старый URL в индексе.
- **SEO:** Вес не передается (риск потери позиций).
- **Кэш:** Не кэшируется (обычно).



**Внимание:** Google со временем может воспринять длительный 302 как 301, но полагаться на это рискованно.

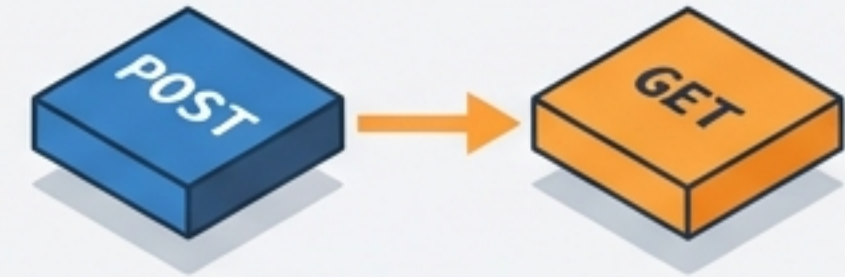
# ПРОДВИНУТЫЕ ТИПЫ: ТОЧНОСТЬ МЕТОДА



**303 (See Other)**

Для форм оплаты.  
Предотвращает повторную  
отправку данных.

Всегда меняет метод запроса на GET.



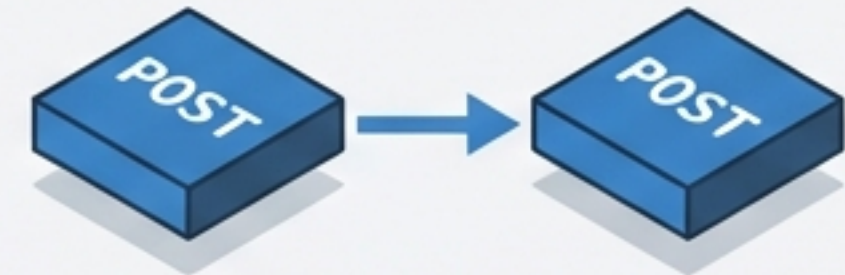
✓ POST → GET



**307 (Temporary)**

Современный аналог 302  
(HTTP 1.1).

Гарантирует сохранение метода  
запроса (POST остается POST).



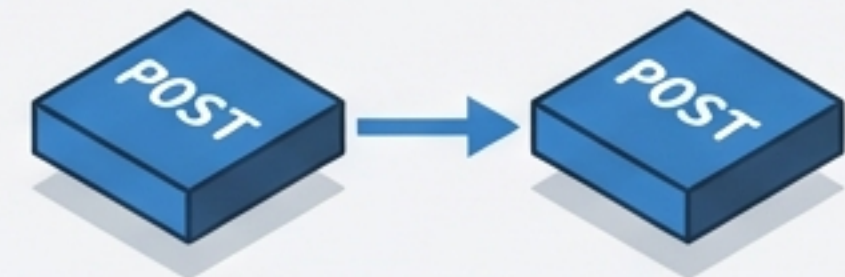
✓ POST → POST



**308 (Permanent)**

Современный аналог 301.

Постоянный переезд. Сохраняет метод POST.  
Экспериментальный, но точный.



✓ POST → POST

# КЛИЕНТСКИЕ РЕДИРЕКТЫ: ОПАСНАЯ ЗОНА

## Meta Refresh

```
<meta http-equiv="refresh"
content="5; url=...">
```

- Очень медленно.
- Раздражает пользователя (отсчет времени).
- Плохо для SEO.




## JavaScript Redirect

```
window.location.replace("...")
```

- ненадежно (зависит от настроек браузера).
- Google может пропустить переход при сканировании.



 **ВЕРДИКТ:** Используйте только если нет доступа к серверным настройкам (.htaccess).

# ТОП-5 СЦЕНАРИЕВ ИСПОЛЬЗОВАНИЯ



## Склейка зеркал

www → non-www  
http → https



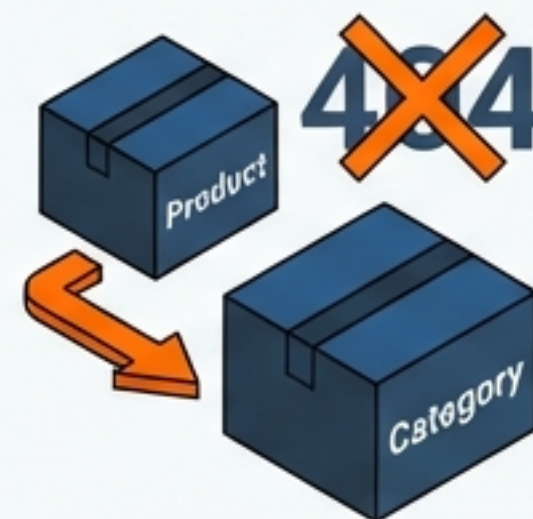
## Trailing Slash

/page/ → /page  
Устранение  
дублей.



## Смена домена

Полный переезд  
бренда с  
сохранением  
трафика.



## Удаление товаров

Редирект на  
категию вместо  
ошибки 404.



## Смена регистра

/Page → /page  
Стандартизация  
URL.

# ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ

## Apache (.htaccess)

```
Redirect 301 /old-page.html  
/new-page.html
```

Самый популярный метод  
для Linux-хостингов.

## Nginx (config)

```
rewrite ^/old-url/$  
http://site.com/new-url/  
permanent;
```

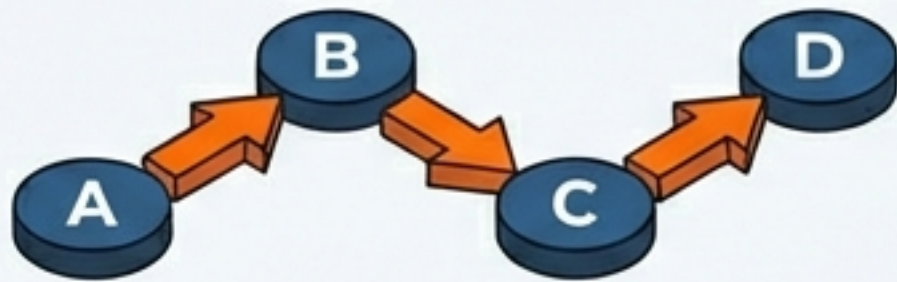
## WordPress



Безопасный вариант  
для новичков (плагины).

# ОПАСНЫЕ ОШИБКИ КОНФИГУРАЦИИ

## Цепочки (Chains)

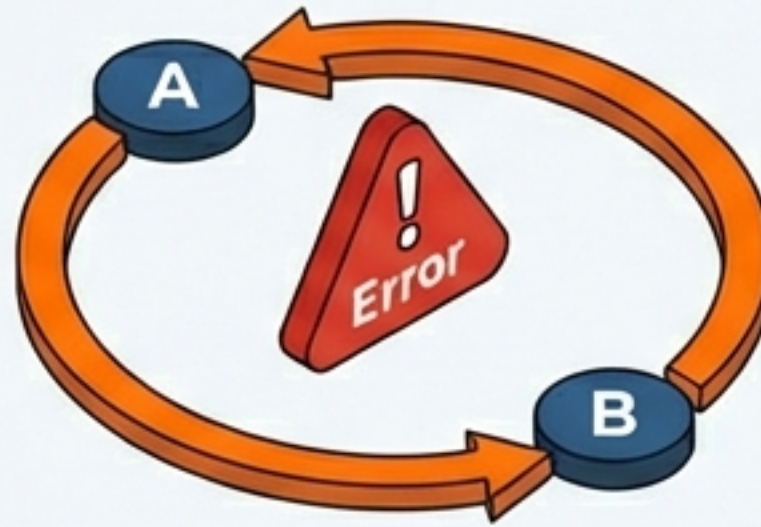


A → B → C → D

→ Правильно: A → D

Теряется вес, растет время загрузки.

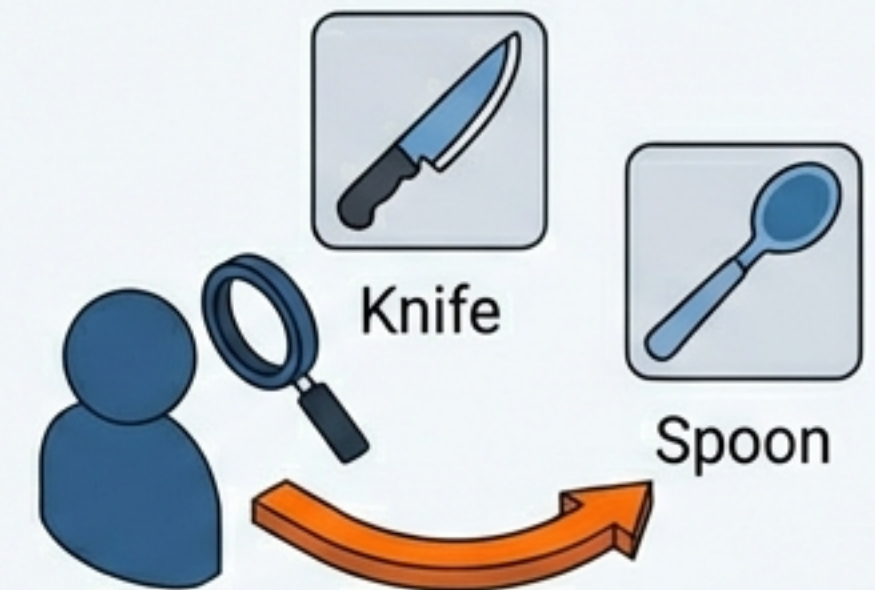
## Петли (Loops)



ERR\_TOO\_MANY\_REDIRECTS

Сайт падает, страница недоступна.

## Soft 404






Редирект на нерелевантную страницу (например, удаленный товар → главная).

# ЧЕК-ЛИСТ: КАКОЙ РЕДИРЕКТ ВЫБРАТЬ?



## Инструменты для проверки:

 Google Search Console  Screaming Frog  Redirect Detective

Всегда проверяйте на цепочки и петли после настройки.