

Dynamic Api's

- 1. cookie/header
- 2. connection/draftMode
- 3. searchParams
- 4. dynamic='force-dynamic'
- 5. cache: 'no-store'
- 6. revalidate = 0

- render types:
- ssg - server site generate
  - ssr - server side rendering
  - csr - client side rendering
  - isr - incremental static rendring

|              | <div> <div></div> <div>Вид</div> </div> | <div> <div></div> <div>Время жизни</div> </div>                         | <div> <div></div> <div>Описание</div> </div>   | <div> <div></div> <div>Инвалидация</div> </div>  | <div> <div></div> <div>+</div> </div> |
|--------------|---|---|--|--|---------------------------------------|
|              | Request Memoization                     | Кеш действует в рамках одного запроса                                   | переиспользует результат fetch из разных компонентов/ generateMetadata/ generateStaticParams. После запрос кеш сбрасывается. Подходит только для GET запросов, если у вас POST (например, graphql), то нужно оборачивать в react cache | отсутствует, так как действует в рамках 1 запроса  |                                       |
|              | Data cache                              | Кеш действует между запросами и между релизами                          | к запросу нужно добавить cache: "force-cache"/revalidate/tags , тогда результат будет закеширован. cache: "no-store" отключает кеширование и запрос будет по сети  | инвалидировать можно только с помощью по времени (revalidate) или по запросу (revalidatePath или revalidateTag)  |                                       |
|              | Full route cache                        | Кешируется между запросами  | <div>Хранение результатов рендера RSC payload и HTML на сервере. Кешируются только статичные сегменты.</div> <div>dynamic = 'force-static', для того, чтобы сделать сегмент статичным или пустой массив из generateStaticParams</div>  | Ивалидируется после редеплоя или инвалидации Data Cache  |                                       |
|              | Client-side router cache                | Действует только в рамках сессии (при обновлении страницы сбрасывается) | Хранит RSC payload на клиенте. Layout, loading states по умолчанию переиспользуются, а страницы перерисовываются (но при навигации вперед/назад по стрелкам не перерисовываются + есть настройка staleTimes)                           | <div>Инвалидация - автоматическая в зависимости от параметры prefetch</div> <div>prefetch={null} or unspecified: not cached for dynamic pages, 5 minutes for static pages.</div> <div>prefetch={true} or router. prefetch: 5 minutes for both static &amp; dynamic pages. И ручная: в server actions revalidatePath или revalidateByTag, cookies.set или cookies.delete</div> <div>Инвалидировать нужно в server actions</div> |                                       |
| <div>+</div> |   |   |  |  |                                       |

server

next create react server components  
payload (RSC payload)

использует RSC payload и результат client  
components -> render html

client (browser)

отображает html

создает дерево, использует rsc payload

гидрация





