??? ????????: options.routes = cloneDeep(options.routes);

Эта строка означает:
□□ создаётся глубокая копия массива маршрутов, и она записывается обратно в `options.routes`.

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

```
options.routes = cloneDeep(options.routes);
```

??? ????? `cloneDeep`

`cloneDeep` — обычно функция из **lodash**:

```
import { cloneDeep } from 'lodash-es';
```

Она:

- создаёт **полную (глубокую) копию** объекта
- копирует **все вложенные объекты и массивы**
- не оставляет ссылок на оригинал

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

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

```
const copy = [...options.routes];
```

Проблема:

- вложенные объекты (`children`, `meta`) остаются **общими по ссылке**

```
copy[0].meta.title = 'New';
// изменится и в original 
```

? `cloneDeep`

```
const copy = cloneDeep(options.routes);
```

Теперь:

```
copy[0].meta.title = 'New';
// original НЕ изменится 
```

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

1 Берётся `options.routes`
2 Создаётся **полный клон** всей структуры маршрутов
3 `options.routes` теперь указывает **на новую копию**
4 Оригинальные маршруты остаются нетронутыми

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

В коде генерации роутов обычно:

- добавляют `children`
- фильтруют по ролям
- меняют `meta`
- добавляют layout

 Если менять оригинал:

- можно сломать повторную генерацию
- получить дубликаты
- испортить базовую конфигурацию

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

```
options.routes = cloneDeep(options.routes);


// дальше
options.routes.forEach(route => {
  route.children?.push(dynamicRoute)
})
```

◫ изменения **безопасны**
◫ исходные `routes` не портятся

???????? ?

- Без `cloneDeep` — ты редактируешь **оригинал**
- С `cloneDeep` — работаешь с **копией документа**

???????

> Эта строка **защищает исходные маршруты**,
> создавая их **полную независимую копию** перед изменениями.

Revision #1
Created 2026-01-20 09:18:09 UTC by Nikolay
Updated 2026-01-20 09:18:51 UTC by Nikolay