Сравнение `useNavigate` и `Outlet` в React Router

1. Назначение

`useNavigate`

- Хук для **программной навигации**.
- Используется для перехода между страницами в ответ на действия пользователя или события (например, клик, логин, сабмит формы).

`Outlet`

- Компонент для **отображения вложенных маршрутов**.
- Служит «точкой рендера», в которую React Router подставляет дочерние компоненты.

---

2. Пример использования `useNavigate`

```
import { useNavigate } from "react-router-dom";

function Login() {
  const navigate = useNavigate();

  const handleLogin = () => {
    // Логика авторизации
    navigate("/dashboard"); // переход программно
  };

  return (
    <div>
      <h1>Вход</h1>
      <button onClick={handleLogin}>Войти</button>
    </div>
  );
}
```

📌Здесь `useNavigate` нужен, потому что переход выполняется **после логики авторизации**.

---

3. Пример использования `Outlet`

```
import { Outlet, Link } from "react-router-dom";

function Layout() {
```

```
  return (
    <div>
      <nav>
        <Link to="/">Главная</Link>
        <Link to="/about">О нас</Link>
      </nav>
      <hr />
      <Outlet /> {/* сюда попадет компонент вложенного маршрута */}
    </div>
  );
}
```

📌Здесь `Outlet` используется для **вложенных маршрутов**. Например, внутри `/` можно отобразить и главную страницу, и страницу «О нас».

---

4. Сравнение

Критерий	`useNavigate`	`Outlet`
Тип	Хук	Компонент
Основная задача	Программная навигация	Отображение вложенных маршрутов
Когда используется	После действий (клик, логин, сабмит)	В layout-компонентах
Альтернатива	`<Link>`, `<NavLink>`	Нет прямой альтернативы
Где применяется	В логике компонентов	В структуре маршрутов

---

5. Итог

- `useNavigate` и `Outlet` выполняют **разные роли**:
- `useNavigate` нужен для **управления навигацией** через код.
- `Outlet` нужен для **организации вложенной структуры маршрутов**.
- 🔩Один не может заменить другой.
- 🔗Они часто используются **вместе**: `Outlet` — для структуры, `useNavigate` — для логики переходов.