



Firestore أبسط استراتيجية لجلب المنتجات من

بناءً على محادثتنا السابقة ورغبتك في الحصول على أبسط حل، إليك الاستراتيجية الأكثر بساطة وفعالية:

الاستراتيجية المبسطة

الفكرة الأساسية:

- `getServerSideProps` صفحة واحدة تجلب المنتجات عبر
- `initialData` للكاش مع `React Query`
- معقد `Context` تمرير مباشر للبيانات بدون
- تحديث كل 60 ثانية تلقائياً

البنية المطلوبة

```
`` fabrica-store/ |--- lib/ |   |--- data.js |--- pages/ | |--- api/ | |   |--- products.js | |--- products/ | | |--- index.js | |   |--- [id].js |   |--- _app.js |--- package.json ``
```

الكود العملي

1) Firebase البيانات إعداد

```
**lib/data.js** ``js import { db } from '../firebase/config'; import { collection, getDocs, doc, getDoc } from 'firebase/firestore'; export async function getAllProducts() { const snap = await getDocs(collection(db, 'products')); return snap.docs.map((d) => ({ id: d.id, ...d.data() })); } export async function getProduct(id) { const ref = doc(db, 'products', id); const snap = await getDoc(ref); return snap.exists() ? { id: snap.id, ...snap.data() } : null; } ``
```

2) API Route بسيطة

```
**pages/api/products.js** ``js import { getAllProducts } from '../lib/data'; export default async function handler(req, res) { try { res.setHeader('Cache-Control', 'public, s-maxage=60'); const products = await getAllProducts(); res.status(200).json(products); } catch (error) { res.status(500).json({ message: 'Failed to load products' }); } } ``
```

3) إعداد React Query

```
**pages/_app.js** ```js import { QueryClient, QueryClientProvider } from '@tanstack/react-query'; import { useState } from 'react'; export default function MyApp({ Component, pageProps }) { const [queryClient] = useState( () => new QueryClient({ defaultOptions: { queries: { staleTime: 60_000, // 60 ثانية refetchOnWindowFocus: false, }, }, }) ); return ( ); } ``
```

4) صفحة قائمة المنتجات

```
**pages/products/index.js** ```js import { useQuery } from '@tanstack/react-query'; import Link from 'next/link'; import { getAllProducts } from '../lib/data'; async function fetchProducts() { const res = await fetch('/api/products'); return res.json(); } export default function ProductsPage({ initialProducts }) { const { data: products = initialProducts } = useQuery({ queryKey: ['products'], queryFn: fetchProducts, initialData: initialProducts, }); return (
```

المنتجات

```
{products.map((product) => (
```

```
{product.name_ar}
```

```
{product.price} جنيه
```

```
    ))}
</div>
```

```
);
}
```

```
export async function getServerSideProps() {
  const products = await getAllProducts();
  return {
    props: { initialProducts: products },
  };
}
```

```
</div>
```

```
### 5) صفحة تفاصيل المنتج
```

```
<div dir="ltr" style="text-align: left;">
```

```
**pages/products/[id].js**
```

```
```js
```

```
import { getProduct, getAllProducts } from '../lib/data';
```

```
export default function ProductPage({ product }) {
```

```
 if (!product) {
```

```
 return <div>المنتج غير موجود</div>;
```

```

 }

 return (
 <div className="container mx-auto p-4">
 <h1 className="text-3xl font-bold mb-4">{product.name_ar}</h1>
 <p className="text-gray-700 mb-4">{product.description_ar}</p>
 <div className="text-2xl font-bold text-blue-600">{product.price} جنيه</div>
 </div>
);
 }

 export async function getServerSideProps({ params }) {
 const product = await getProduct(params.id);

 return {
 props: { product },
 notFound: !product,
 };
 }
}

```

</div>

## المزايا

### ✔ بساطة قصوى:

- لا Context مع initialData معقد - فقط لا React Query
- لا Hydration معقدة - مباشرة getServerSideProps يمرر البيانات
- لا Route Handlers بسيطة API فقط -

### ✔ أداء ممتاز:

- حقيقي للـ SSR
- كاش 60 ثانية تلقائي
- لكل دقيقة Firestore تحميل واحد من

### ✔ سهولة الصيانة:

- ملفات قليلة وواضحة
- لا تعقيد في البنية
- سهولة الإضافة عليها لاحقاً

## ❑ تثبيت المتطلبات

```
```bash npm install @tanstack/react-query firebase ```
```

❑ النتيجة

هذه الاستراتيجية تحقق 90% من الأهداف بـ 10% من التعقيد

- كل دقيقة Firestore جلب واحد من
- للسير SSR
- سلسلة SPA تجربة
- كود بسيط وقابل للفهم

❑ هل تريد البدء بتطبيق هذه الاستراتيجية؟