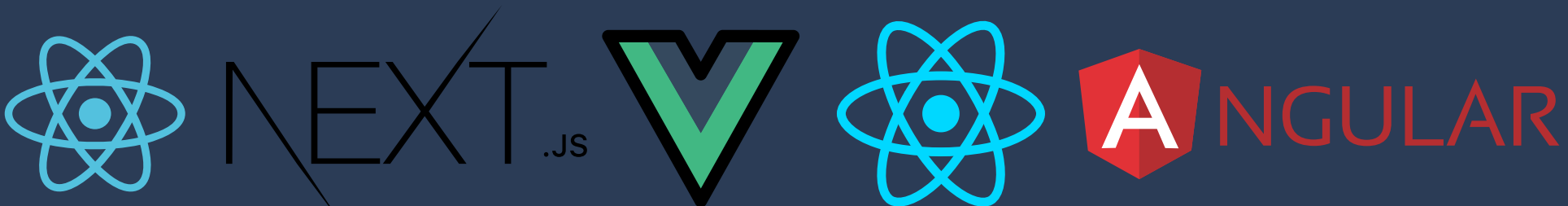
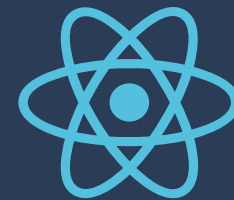


# Master API *Data Fetching* Like a Pro!





Whether you're building with **React**, **Next.js**, **React Native**,  
**Angular**, or **Vue**, we've got the perfect tools and strategies to  
supercharge your data layer.



# FOR REACT



## Best library: TanStack Query

1.  **fetch** API: Simple, native, no dependencies
2.  **axios**: Easy syntax, supports interceptors & base URLs
3.  **TanStack React Query**: Powerful caching, pagination, retry, background sync
4.  **SWR** (by Vercel): Stale-while-revalidate, auto-fetch on focus

 **Best Use**: Dynamic UIs, dashboards, real-time apps

## FOR NEXT.JS

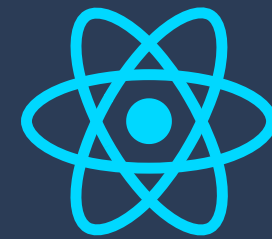


Best combo





**getStaticProps + SWR or React Query**

1. **getServerSideProps**: SSR, fetch on every request
  2. **getStaticProps**: SSG, builds at compile time
  3. **Client-side Fetching**: Using fetch or axios in useEffect
  4. **SWR**: Lightweight client fetching with cache + auto updates
  5. **React Query**: Full data layer control in large apps
- Best Use**: SEO-focused pages, hybrid rendering

# FOR REACT NATIVE



## **Best combo : Axios + React Query**

1.  **fetch**: Native to RN, works out of the box
2.  **axios**: Cleaner API, handles timeouts, headers, etc.
3.  **React Query**: Offline caching, retry, background updates
4.  **REST or GraphQL Compatible**: Easy to hook into any API






 **Best Use**: Mobile apps needing state sync with APIs

# FOR ANGULAR



 **Best combo**

## **HttpClient + RxJS + NgRx (if needed)**

1.  **HttpClientModule**: Angular's built-in HTTP tool
2.  **Observables + RxJS**: Reactive, cancellable, composable
3.  **Interceptors**: Modify requests globally (e.g., auth headers)
4.  **NgRx/Signal Store**: State management with effects
5.  **Async Pipe**: Automatically subscribe/unsubscribe



 **Best Use**: Enterprise-scale apps with RxJS streams

# FOR VUE.JS / NUXT



 **Best combo**

**Axios + TanStack Vue Query or Nuxt features**

1.  **axios**: Most common, easy to use
2.  **@tanstack/vue-query**: Caching, stale revalidation, loading states
3.  **Composition API**: Clean integration with async/await
4.  **Nuxt asyncData / fetch**: Server-side data fetching built-in
5.  **Pinia + Axios**: Lightweight alternative to Vuex

 **Best Use:** SPAs, SSR apps with dynamic content

## Summary

### *What to pick?*

Framework/Library	Recommended Methods
React	fetch, axios, react-query, SWR
Next.js	getServerSideProps, getStaticProps, SWR, axios
React Native	axios, fetch, react-query
Angular	Angular HttpClient, RxJS, NgRx
Vue	axios, @tanstack/vue-query, Nuxt features



# Ali Asghar

## SOFTWARE ENGINEER



/AliAsgharSWE



/AliAsgharSWE



aliasghar.softeng.tech



aliasghar@softeng.tech

aliasghar.softeng.tech