Bluesky App Project Structure

The official client for the Bluesky social network. A cross-platform React Native (Expo) application that runs on iOS, Android, and the Web.

Updated 2025-12-30

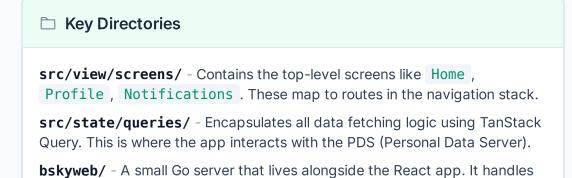#bluesky #react-native #expo #typescript #at-protocol #social-media

PNG   PDF   Copy   </> Prompt

Project Directory

```
social-app/
  src/          Main Applicatio…
    view/       UI Components
      screens/  Page-level views
      com/      Shared componen…
      shell/    App layout/nav
    state/      Application Sta…
      queries/  React Query hoo…
      models/   Data models
    lib/        Utilities
      api/      AT Protocol cli…
  modules/      Native Modules …
  bskyweb/      Go server for S…
  app.config.js Expo config
  package.json  Dependencies
```

Repository Info

Repository - bluesky-social/social-app

Stars - 15k+

License - MIT

Last Analyzed - December 2025

Tech Stack

Framework - React Native + Expo

Web - React Native Web

Language - TypeScript

Protocol - AT Protocol

State - React Query + MobX (State tree)

Styling - StyleSheet (React Native)

Architecture Notes

Bluesky is a 'Universal' React Native app. It uses the same codebase to render the iOS/Android apps and the `bsky.app` web interface. The `src/view` directory contains the UI, largely built with primitives that map to native views on mobile and HTML on web. State management handles the complexity of caching social graph data from the AT Protocol.

Key Directories

`src/view/screens/` - Contains the top-level screens like `Home`, `Profile`, `Notifications`. These map to routes in the navigation stack.

`src/state/queries/` - Encapsulates all data fetching logic using TanStack Query. This is where the app interacts with the PDS (Personal Data Server).

`bskyweb/` - A small Go server that lives alongside the React app. It handles server-side rendering of meta tags for SEO and link previews (OpenGraph), which a client-side SPA cannot do easily.

Why This Structure?

Bluesky is one of the most prominent open-source React Native Web applications. It proves that you can build a high-quality, performant social network client that shares 99% of code between mobile and web.