

Pokemon List with Caching and Favourites

Objective:

Build an iOS app that displays a list of Pokemons from the [PokeAPI](https://pokeapi.co/), supports pagination, and allows users to add Pokemons to a favourites list. The app should cache data locally, ensuring a fast and seamless user experience.

Project Requirements:

- **API:** [PokeAPI](https://pokeapi.co/)
- **Endpoint:** <https://pokeapi.co/api/v2/pokemon?limit=10&offset=0>
- **Image URL:** sprites -> other -> official-artwork -> front_default
- **Platform:** iOS 17+
- **Device:** iPhone only
- **Orientation:** Portrait only
- **Theme:** Light + Dark mode
- **Locale:** EN only
- **Framework:** UIKit or SwiftUI (Developer's choice)
- **Third-Party Libraries:** Allowed (ANY)
- **Distribution:** Archived Xcode project
- **Time to Develop:** ~1 hour
- **App architecture:** Any
- **No constraints on architecture**
- **Creativity and Proactiveness are encouraged:** Extra UI/UX improvements or optimizations are welcomed.

Tasks:

1. First Screen - List of Pokemons:

- Display a list of Pokemons
(<https://pokeapi.co/api/v2/pokemon?limit=10&offset=0>)
- For each cell, show:
 - **Name** of the Pokemon
 - **Image** from sprites -> other -> official-artwork -> front_default
- Add **Pull-to-Refresh** functionality to reload the list.
- Use loaders/spinners where necessary for a smooth loading experience.

2. Add Pagination:

- Implement pagination to load more Pokemons while scrolling.
- Adjust the offset dynamically as the user scrolls.

3. Detail Screen:

- On tapping a Pokemon, navigate to a **Detail Screen**.
- The detail screen can be presented or pushed (Developer's choice).
- Display the following details:
 - **Name**
 - **Image**
 - **Experience**
 - **Height**
 - **Weight**

4. Favourites Logic:

- Add a **star/star.fill SF Symbol** icon to each Pokemon cell to mark it as favourite.
- Tapping the star icon should:
 - Toggle between favourite and non-favourite state.
 - Persist the favourite state, even after app restarts.

5. Caching:

- Cache Pokemon data locally so that the list loads instantly when reopening the app.
- Use any preferred caching mechanism

6. Segmented Control for All/Favourites:

- Add a **Segmented Control** at the top of the First Screen.
- Options: **All | Favourites**
- Toggling between the segments should:
 - Display all Pokemons or only the favourite ones.

7. Toolbar for Favourites Management:

- When **Favourites** segment is selected:
- Display a toolbar with a **Trash (SF Symbol)** icon.
- On tapping the Trash icon:
 - Show a **confirmation alert** to delete all favourites.
 - If confirmed, clear all favourites and update the UI.

8. Enhanced UI/UX:

- Show a subtle, non-blocking loading indicator at the bottom while loading new pages.
- Maintain smooth scrolling without blocking the UI.
- Implement empty state UI for:
 - No favourites added.

9. Error Handling and Offline Mode:

- Display error messages for failed network requests.
- Include a retry button for failed requests.
- Support offline mode by loading cached data when no internet connection is available.