

```

//
// DriversView.swift
// RaceReady
//
// Created by Adnann Muratovic on 13.07.24.
//

import SwiftUI

struct WidthPreferenceKey: PreferenceKey {
    static var defaultValue: CGFloat = 0
    static func reduce(value: inout CGFloat, nextValue: () -> CGFloat) {
        value = nextValue()
    }
}

struct DriversView: View {
    @EnvironmentObject var driverStandingsModel: DriverStandingsModel

    var body: some View {
        VStack {
            Group {
                if driverStandingsModel.isLoading {
                    ProgressView("Loading drivers...")
                } else if let errorMessage =
                    driverStandingsModel.errorMessage {
                    VStack {
                        Text("Error: \(errorMessage)")
                            .foregroundColor(.red)
                        if driverStandingsModel.isCacheAvailable {
                            Text("Showing cached data")
                                .foregroundColor(.orange)
                        } else {
                            Text("No cached data available")
                                .foregroundColor(.orange)
                        }
                    }
                    Button("Retry") {
                        Task {
                            await
                                driverStandingsModel
                                    .populateDriverStandings()
                        }
                    }
                    .padding()
                    .background(Color.blue)
                    .foregroundColor(.white)
                    .cornerRadius(8)
                }
                .padding()
            } else if driverStandingsModel.driverStandings.isEmpty {
                Text("No driver standings available")
                    .foregroundColor(.secondary)
            } else {
                List {

```

```

        ForEach(driverStandingsModel.driverStandings, id:
            \.driver.driverId) { driverStanding in
                DriverStandingView(driverStanding:
                    driverStanding)
            }
        }
    }

VStack {
    if let lastUpdated = driverStandingsModel.lastUpdated {
        Text("Last updated: \((formattedDate(lastUpdated))")
    }
    Text("Source: \((driverStandingsModel.dataSource == .api ?
        "API" : "Cache")")
        .foregroundColor(driverStandingsModel.dataSource ==
            .api ? .green : .orange)
    if driverStandingsModel.isCacheAvailable {
        Text("Cache available")
            .foregroundColor(.blue)
    } else {
        Text("No cache available")
            .foregroundColor(.red)
    }
}
.font(.caption)
.padding()
.frame(maxWidth: .infinity)
.background(Color.gray.opacity(0.1))
}
.navigationTitle("F1 Drivers 2024")
.navigationBarTitleDisplayMode(.large)
.onAppear {
    if driverStandingsModel.driverStandings.isEmpty {
        Task {
            await driverStandingsModel.populateDriverStandings()
        }
    }
}

private func formattedDate(_ date: Date) -> String {
    let formatter = DateFormatter()
    formatter.dateStyle = .medium
    formatter.timeStyle = .short
    return formatter.string(from: date)
}

#Preview {
    NavigationStack {
        DriversView()
            .environmentObject(DriverStandingsModel(webService:
                WebService()))
    }
}

```

