Network basics

Simple network calls for SwiftUI

Network basics

- Cleartext HTTP calls are blocked by iOS
  - HTTPS is mandatory, or we need to turn off App Transport Security
- Network calls are handled by the URLSession class
- JSON encoding/decoding can be done using the Codable protocol

Codable

Encodable & Decodable

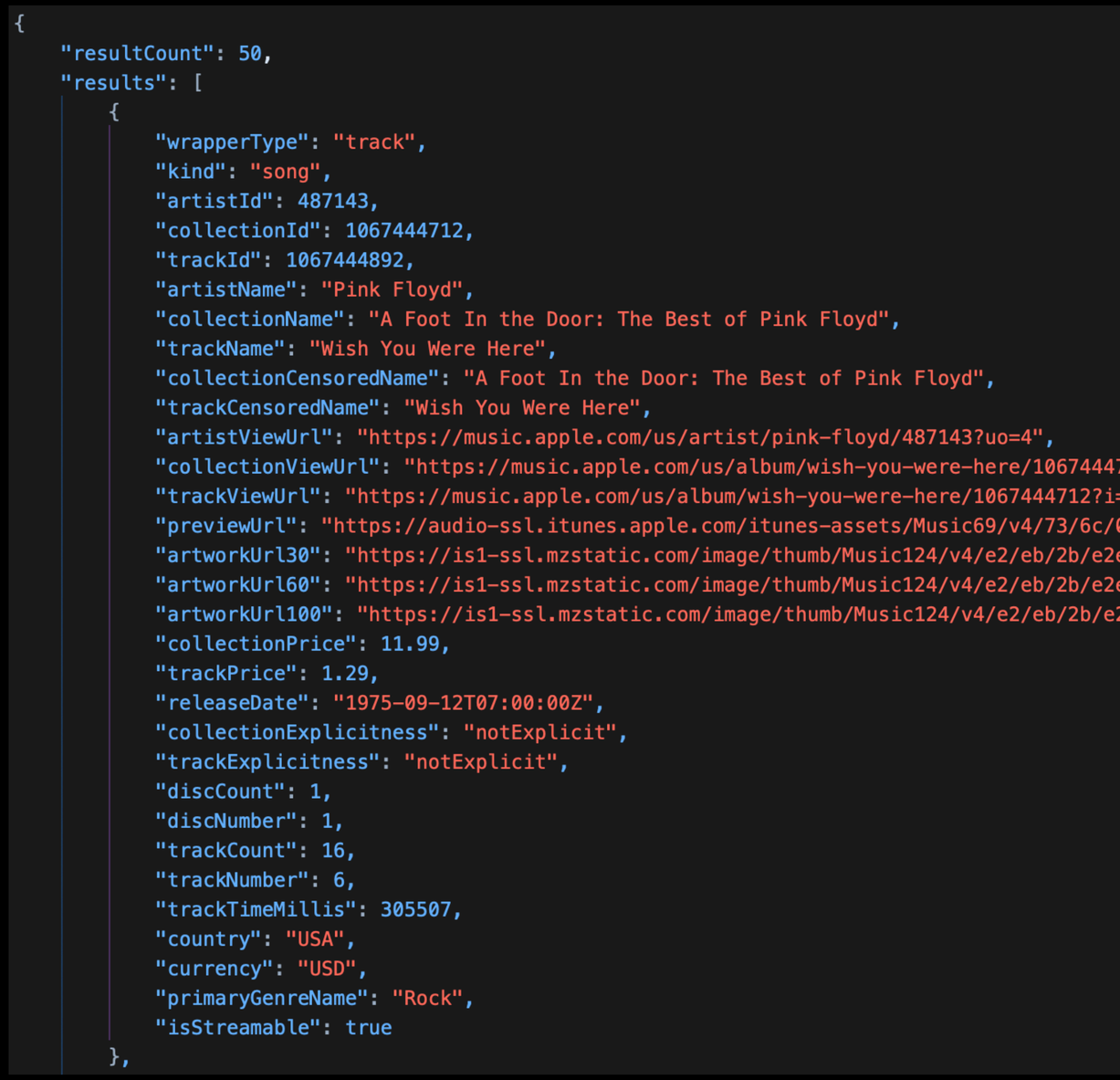
- Built-in way the encode / decode Swift types to other format
  - JSON and Property List are provided by default
  - We can build support for other formats
- Swift can auto-implement the protocol on a type automatically if all properties conforms to Codable

```
struct Song: Codable {
    var trackId: Int
    var trackName: String
    var artistName: String
    var previewUrl: URL
}

struct ITMSResponse: Codable {
    var results: [Song]
}
```

```swift
struct Song: Codable {
    var trackId: Int
    var trackName: String
    var artistName: String
    var previewUrl: URL
}

struct ITMSResponse: Codable {
    var results: [Song]
}
```




```
{
    "resultCount": 50,
    "results": [
        {
            "wrapperType": "track",
            "kind": "song",
            "artistId": 487143,
            "collectionId": 1067444712,
            "trackId": 1067444892,
            "artistName": "Pink Floyd",
            "collectionName": "A Foot In the Door: The Best of Pink Floyd",
            "trackName": "Wish You Were Here",
            "collectionCensoredName": "A Foot In the Door: The Best of Pink Floyd",
            "trackCensoredName": "Wish You Were Here",
            "artistViewUrl": "https://music.apple.com/us/artist/pink-floyd/487143?uo=4",
            "collectionViewUrl": "https://music.apple.com/us/album/wish-you-were-here/1067444
            "trackViewUrl": "https://music.apple.com/us/album/wish-you-were-here/1067444712?i
            "previewUrl": "https://audio-ssl.itunes.apple.com/itunes-assets/Music69/v4/73/6c/
            "artworkUrl30": "https://is1-ssl.mzstatic.com/image/thumb/Music124/v4/e2/eb/2b/e2
            "artworkUrl60": "https://is1-ssl.mzstatic.com/image/thumb/Music124/v4/e2/eb/2b/e2
            "artworkUrl100": "https://is1-ssl.mzstatic.com/image/thumb/Music124/v4/e2/eb/2b/e
            "collectionPrice": 11.99,
            "trackPrice": 1.29,
            "releaseDate": "1975-09-12T07:00:00Z",
            "collectionExplicitness": "notExplicit",
            "trackExplicitness": "notExplicit",
            "discCount": 1,
            "discNumber": 1,
            "trackCount": 16,
            "trackNumber": 6,
            "trackTimeMillis": 305507,
            "country": "USA",
            "currency": "USD",
            "primaryGenreName": "Rock",
            "isStreamable": true
        },
```

```swift
func loadData() {

    guard let url = URL(string: "https://itunes.apple.com/search?term=pink+floyd&entity=song") else { return }
    let request = URLRequest(url: url)
    URLSession.shared.dataTask(with: request) { data, response, error in
        // Handle network response
    }.resume()

}
```

```swift
func loadData() {

    guard let url = URL(string: "https://itunes.apple.com/search?term=pink+floyd&entity=song") else { return }
    let request = URLRequest(url: url)
    URLSession.shared.dataTask(with: request) { data, response, error in
        // Handle network response
    }.resume()

}
```

```swift
func loadData() {
    guard let url = URL(string: "https://itunes.apple.com/search?term=pink+floyd&entity=song") else { return }
    let request = URLRequest(url: url)
    URLSession.shared.dataTask(with: request) { data, response, error in

        // Handle network response
        guard let data = data else { return }
        if let decodedResponse = try? JSONDecoder().decode(ITMSResponse.self, from: data) {
            DispatchQueue.main.async {
                self.results = decodedResponse.results
            }
        }

    }.resume()
}
```

SwiftUI integration

- We can use URLSession and Codable to download and parse JSON
- We can call our function when the SwiftUI view appears
  - Not on init, as the view is initialised everytime it changes

```swift
struct ContentView: View {

    @State private var results = [Song]()

    var body: some View {
        List(results, id: \.trackId) { item in
            HStack {
                VStack(alignment: .leading) {
                    Text(item.trackName)
                        .font(.headline)
                    Text(item.artistName)
                }
            }
        }.onAppear {
            loadData()
        }
    }

    func loadData() { }
}
```

```
struct ContentView: View {

    @State private var results = [Song]()

    var body: some View {
        List(results, id: \.trackId) { item in
            HStack {
                VStack(alignment: .leading) {
                    Text(item.trackName)
                        .font(.headline)
                    Text(item.artistName)
                }
            }
        }.onAppear {
            loadData()
        }
    }

    func loadData() { }
}
```