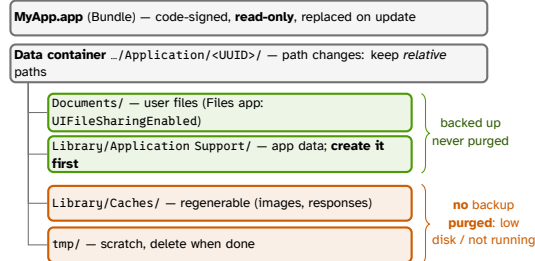


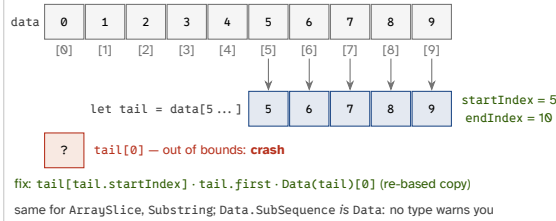
In one line: Most Foundation bugs come from doing by hand what a type already does correctly: build URLs with **URLComponents**, format numbers with a **formatter**/.formatted, convert with **Measurement** — and respect the **sandbox lifecycle**: Caches and tmp are purgeable, the container path changes, Data slices keep their parent's indices.

the sandbox — who survives what

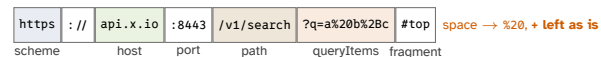


big re-downloadable data kept in App Support → set `isExcludedFromBackup`

Data slice keeps the parent's indices



URLComponents — one property per part



How it works

- URL** = parsed, immutable. `URL(string:)` is nil on invalid input (iOS 17+: percent-encodes it instead, RFC 3986 — differs by OS). Files: `URL(filePath:)` (iOS 16), not `URL(string: "/p")`.
- URLComponents** = builder: set scheme/host/path/queryItems, read `.url`; never interpolate input into a URL string. `percentEncodedQueryItems` (iOS 11) when you encode yourself.
- Data** — value type, COW, Hashable by content; base64 built in, hex not. `Data(contentsOf:)` is **synchronous** — files only, never network. Write with `[.atomic, .completeFileProtection]`.
- Directories** (iOS 16 statics) — `URL.documentsDirectory`, `.applicationSupportDirectory`, `.cachesDirectory`, `.temporaryDirectory`; before: `FileManager.urls(for:in:)`. App Support may not exist yet: `createDirectory`. Extensions share an App Group container.
- NSCache** — thread-safe; evicts under memory pressure; limits are *hints*; keys/values must be **classes**; keys *not copied*; not enumerable, not Codable — a memory tier, never storage. Dictionary: deterministic, value keys, but unbounded and needs a lock/actor.
- Measurement** — `Measurement(value: 5, unit: UnitLength.kilometers).converted(to: .miles).formatted(width: .abbreviated, usage: .road)` picks km/mi by locale.
- Numbers** — money is `Decimal`; `.formatted(.currency(code: "EUR"))`; `0.25.formatted(.percent)` → "25%". Locale decides *layout*, not *which* currency — set `currencyCode`.
- UUID** — v4, 122 random bits: idempotency keys, local ids; `uuidString` is **uppercase**. `identifierForVendor` resets when the vendor's last app goes.
- Bundle** — read-only; SPM: `Bundle.module`; test resources: `Bundle(for: Self.self)`, not `.main` (the host app).
- NotificationCenter vs KVO** — broadcast by name vs one `@objc` dynamic property; keep and remove the token either way.

Example

```
var c = URLComponents(string: "https://api.example.com/search")!
c.queryItems = [URLQueryItem(name: "q", value: "a b+c")]
c.percentEncodedQuery = c.percentEncodedQuery? // form servers
  .replacingOccurrences(of: "+", with: "%2B") // read + as space
let url = c.url! // ... search?q=a%20b%2Bc

let tail = data[5...] // startIndex 5, not 0
let head = tail[tail.startIndex] // or Data(tail)[0]

var dir = URL.applicationSupportDirectory // iOS 16
var rv = URLResourceValues(); rv.isExcludedFromBackup = true
try dir.setResourceValues(rv) // needs var
let price = Decimal(string: "19.99")! // not 19.99
price.formatted(.currency(code: "EUR")) // "19,99 €" in de_DE
```

Interview traps

- + in a query value (base64, phone numbers) — `URLComponents` leaves it; a form-decoding server reads a **space**. Encode it as `%2B`.
- `slice[0]` on `Data/ArraySlice` — indices are the parent's.
- Storing an absolute file path — the container UUID changes on update/restore.
- Images “vanish” from Caches/ — that is the contract; persistent data belongs in Application Support.
- `Decimal(19.99)` goes through `Double`: already inexact; use `Decimal(string:)` or integer minor units.
- `numberStyle = .currency` alone shows \$ for a euro amount on a US phone — bind `currencyCode`.
- Relying on `NSCache` keeping a value or honouring `countLimit` exactly.

Remember

Build, don't concatenate. Slices keep indices. Caches and tmp are borrowed. Money is Decimal, currency is explicit.

Likely questions

- NSCache vs Dictionary? — auto-evicting + thread-safe vs deterministic + unbounded.
- Downloaded images go where? — Caches on disk, NSCache in memory.
- Show a price? — `Decimal + .currency(code:)`.
- Stable device id? — none; `identifierForVendor`, or your own UUID in Keychain.