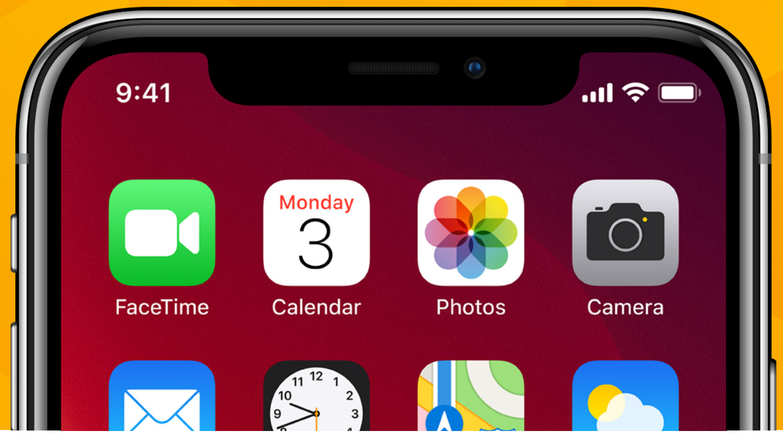


iOS Application Development

Lecture 7: Saving Data • Closures • Animations • Working with the Web

Prof. Dr. Jan Borchers
Media Computing Group
RWTH Aachen University

WS ’22/’23 • hci.rwth-aachen.de/ios

RWTH AACHEN UNIVERSITY

Saving Data

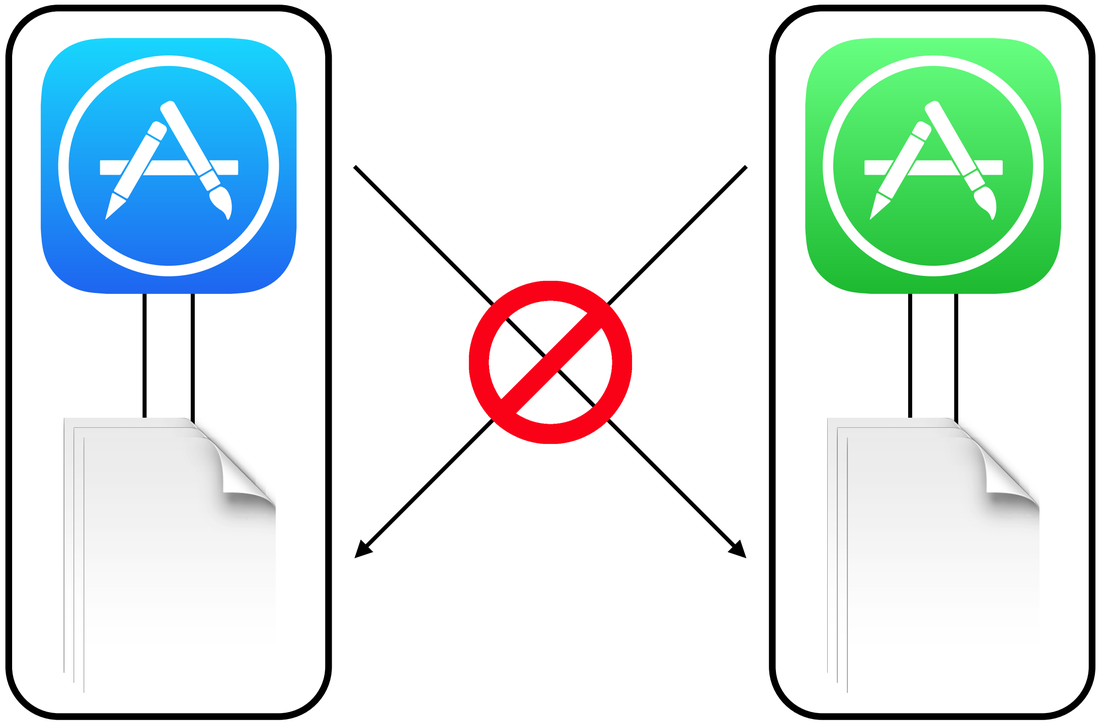
Encoding and Decoding With Codable

- Use Encoder & Decoder objects to encode/decode `Codable` objects
  - `Codable` ↔ `Data`

```
struct Note: Codable {
    let title: String
    let text: String
    let timestamp: Date
}

let myNote = Note(title: "Dry cleaning", text: "Pick up suit from dry cleaners", timestamp: Date())

let propertyListEncoder = PropertyListEncoder()
if let encodedNote = try? propertyListEncoder.encode(myNote) { print(encodedNote) }
```

```
let propertyListDecoder = PropertyListDecoder()
if let decodedNote = try? propertyListDecoder.decode(Note.self, from: encodedNote) { print(decodedNote) }
```

Storing Data: Sandboxing & the FileManager



```
let documentsDirectory =
    FileManager.default.urls(for: .documentDirectory,
        in: .userDomainMask).first!
let archiveURL =
    documentsDirectory.appendingPathComponent("notesData")
        .appendingPathExtension("plist")
```

FileManager: Writing and Loading

```
let documentsDirectory =
    FileManager.default.urls(for: .documentDirectory,
       in: .userDomainMask).first!
let archiveURL =
    documentsDirectory.appendingPathComponent("notesData")
       .appendingPathExtension("plist")
```

Writing

```
let propertyListEncoder = PropertyListEncoder()
let encodedNote = try? propertyListEncoder.encode(myNote)
try? encodedNote?.write(to: archiveURL, options: .noFileProtection)
```

Loading

```
let propertyListDecoder = PropertyListDecoder()
if let retrievedNoteData = try? Data(contentsOf: archiveURL),
    let decodedNote = try? propertyListDecoder.decode(Note.self, from: retrievedNoteData) { print(decodedNote) }
```

Closures

Syntax

```
func sum(numbers: [Int]) -> Int {
  // Code that adds together the numbers array
  return total
}
```

```
let sumClosure = { (numbers: [Int]) -> Int in
  // Code that adds together the numbers array
  return total
}
```

Syntax

```
// A closure with no parameters and no return value
let printClosure = { () -> Void in
  print("This closure does not take any parameters and does not return a value.")
}

// A closure with parameters and no return value
let printClosure = { (string: String) -> Void in
  print(string)
}

// A closure with parameters and a return value
let randomNumberClosure = { (minValue: Int, maxValue: Int) -> Int in
  // Code that returns a random number between `minValue` and `maxValue`
}
```

Passing Closures as Arguments

```
let sortedTracks = tracks.sorted(by: { (firstTrack: Track, secondTrack: Track) -> Bool in
  return firstTrack.trackNumber < secondTrack.trackNumber
})
```

```
let sortedTracks = tracks.sorted { (firstTrack: Track, secondTrack: Track) -> Bool in
  return firstTrack.trackNumber < secondTrack.trackNumber
} // trailing closure syntax!
```

```
let sortedTracks = tracks.sorted { (firstTrack, secondTrack) in
  return firstTrack.trackNumber < secondTrack.trackNumber
}
```

```
let sortedTracks = tracks.sorted { return $0.trackNumber < $1.trackNumber }
```

```
let sortedTracks = tracks.sorted { $0.trackNumber < $1.trackNumber }
```

```
let sortedTracks = tracks.sorted(by: <)
```

Collection Functions Using Closures

```
// Initial array
let firstNames = ["Johnny", "Nellie", "Aaron", "Rachel"]

// Creates an empty array that will be used to store the full names
var fullNames: [String] = []

for name in firstNames {
    let fullName = name + " Smith"
    fullNames.append(fullName)
}
```

```
let fullNames = firstNames.map { (name) -> String in
    return name + " Smith"
}
```

```
let fullNames = firstNames.map { $0 + " Smith" }
```

Collection Functions Using Closures

```
let numbers = [4, 8, 15, 16, 23, 42]

var numbersLessThan20: [Int] = []

for number in numbers {
    if number < 20 {
        numbersLessThan20.append(number)
    }
}
```

```
let numbersLessThan20 = numbers.filter { (number) -> Bool in
    return number < 20
}
```

```
let numbersLessThan20 = numbers.filter { $0 < 20 }
```

Collection Functions Using Closures

```
let numbers = [4, 8, 15, 16, 23, 42]

var total = 0

for number in numbers {
    total = total + number
}
```

```
let total = numbers.reduce(0) { (currentTotal, newValue) -> Int in
    return currentTotal + newValue
}
```

```
let total = numbers.reduce(0, { $0 + $1 })
```

```
let total = numbers.reduce(0, +)
```

Closures Capture Their Environment

- Anything in scope when the closure is created is in scope inside the closure

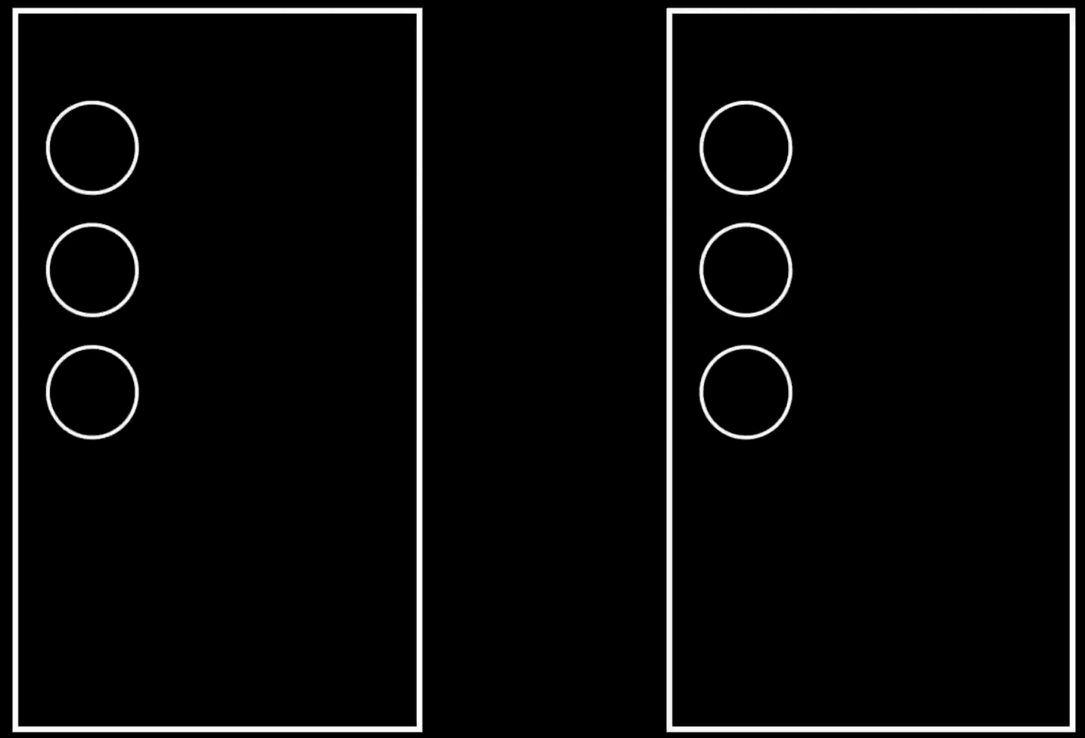
```
animate {
  self.view.backgroundColor = .red
}
```

- If view is removed during animation, view is held in memory until the closure finishes

Animations

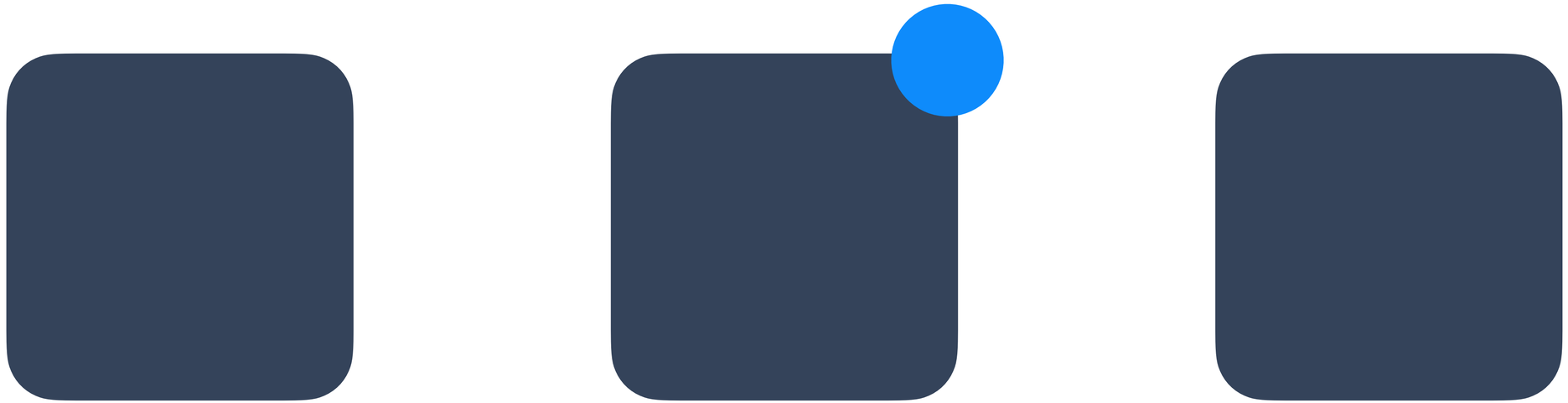
Animations

- Transition from one state to another
- Can define the character of an app



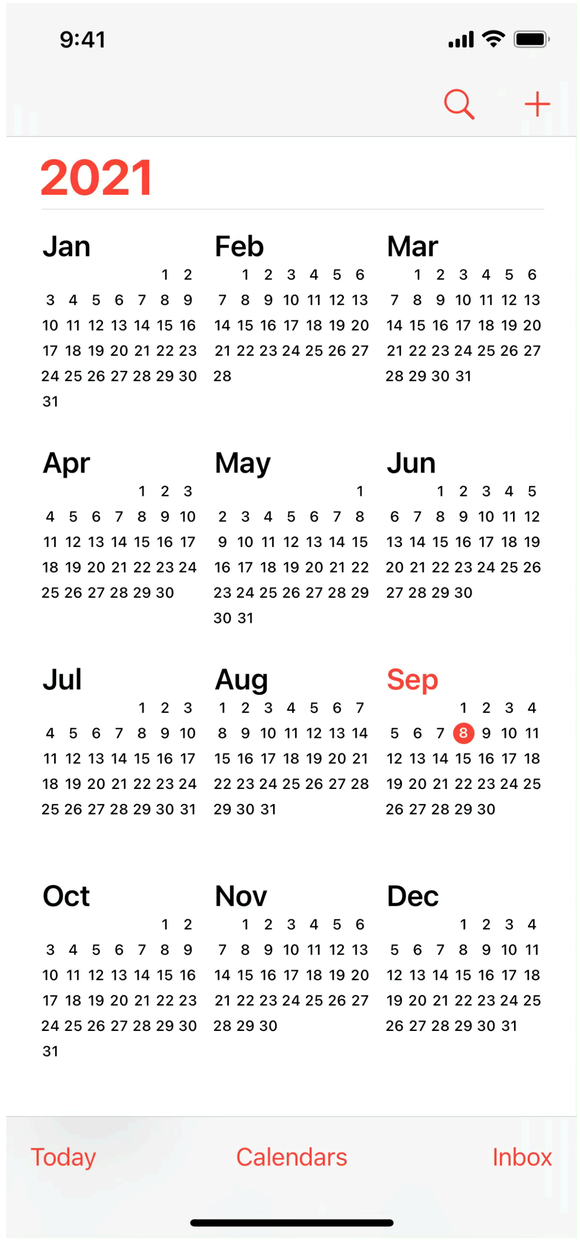
Animations

- Transition from one state to another
- Can define the character of an app
- Direct the user’s attention



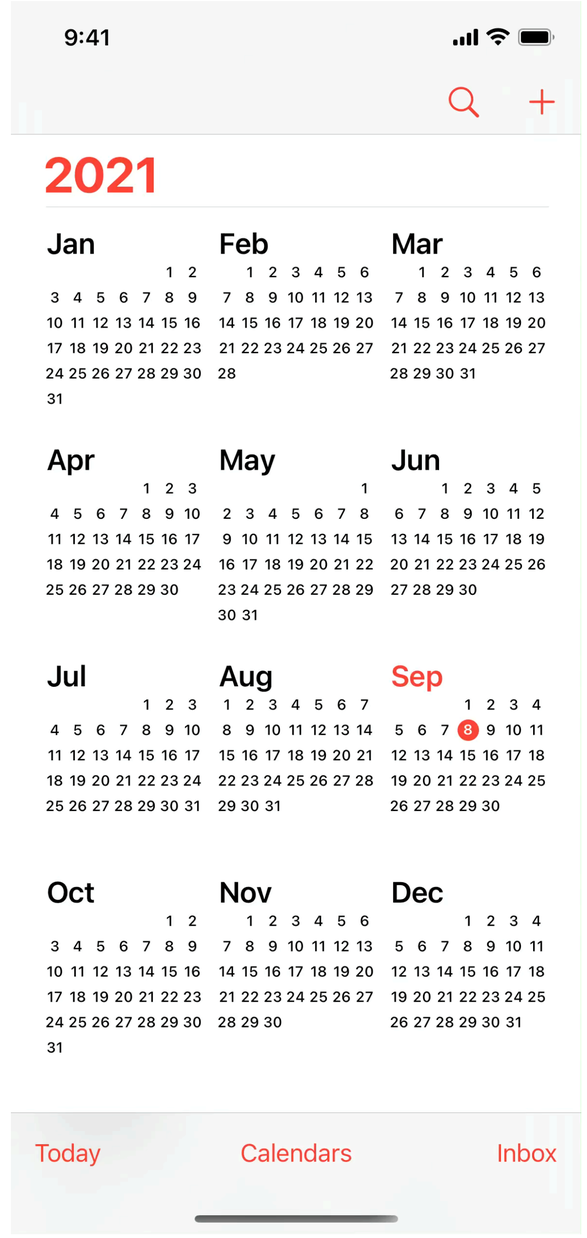
Animations

- Transition from one state to another
- Can define the character of an app
- Direct the user’s attention
- Keep the user oriented



Animations

- Transition from one state to another
- Can define the character of an app
- Direct the user’s attention
- Keep the user oriented
- Connect to user behaviors



What Can Be Animated?

- UIView:
  - alpha
  - backgroundColor
  - bounds
  - center
  - frame
  - transform

UIView Animation Methods

```
animate(withDuration:animations:)
animate(withDuration:animations:completion:)
animate(withDuration:delay:options:animations:completion:)
animate(withDuration:delay:usingSpringWithDamping:initialSpringVelocity:options:animations:completion:)
```

```
UIView.animate(withDuration: 2.0) {
    //animation closure
    view.alpha = 0.3
}
```

UIView Animation Methods

```
animate(withDuration:animations:)
animate(withDuration:animations:completion:)
animate(withDuration:delay:options:animations:completion:)
animate(withDuration:delay:usingSpringWithDamping:initialSpringVelocity:options:animations:completion:)
```

```swift
UIView.animate(withDuration: 1.0, animations: {
    //animation closure
    view.alpha = 1.0
}) { (_: Bool) in
    //completion closure
    UIView.animate(withDuration: 1.0) {
        //second animation closure
        view.alpha = 0.0
    }
}
```

The Transform Property

- Type: `CGAffineTransform`

Type	Initializer	Parameter Description
Scale	`init(scaleX: CGFloat, y: CGFloat)`	The factors by which to scale your view
Rotate	`init(rotationAngle: CGFloat)`	The angle (in radians) by which to rotate your view. Positive value = counterclockwise
Translate	`init(translationX: CGFloat, y: CGFloat)`	The value by which to move (shift) your view

- You can combine transform objects:

```
let scaleTransform = CGAffineTransform(scaleX: 2.0, y: 2.0)
let rotateTransform = CGAffineTransform(rotationAngle: .pi)
let combinedTransform = scaleTransform.concatenating(rotateTransform)
```

Animation in Practice

- Use animation and motion effects judiciously
- Strive for realism and credibility
- Use consistent animation
- Make animations optional

Working With the Web

Working With the Web: What's in a Request?

- URL: https://itunes.apple.com/us/app/keynote/id409183694?mt=12

Protocol	Subdomain	Domain	Path	Query

- Request Type (most common are GET and POST)
- Headers (e.g., User-Agent, authentication, cookies)
- Body (HTML, CSS, images, JSON,…)

Create a URL

- URL:

```
let url = URL(string: "https://www.apple.com")!
```

- Create a data task and the request:

```
Task {
    let (data, response) = try await URLSession.shared.data(from: url)
    if let httpResponse = response as? HTTPURLResponse,
       httpResponse.statusCode == 200,
       let string = String(data: data, encoding: .utf8) {
         print(string)
    }
}
```

asynchronous

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-US" lang="en-US" prefix="og: http://ogp.me/ns#" class="no-js" data-layout-name="availability">

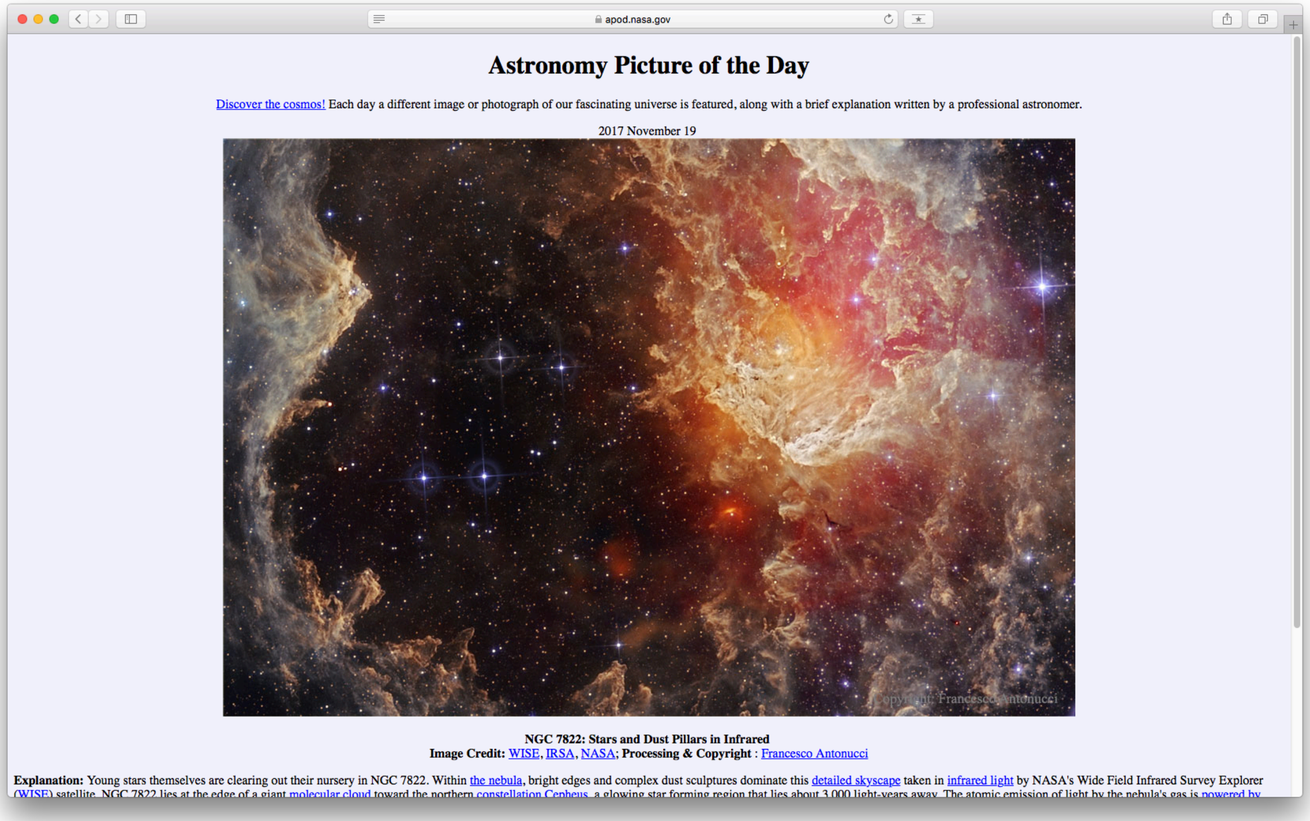
<head>

<meta charset="utf-8" />

<link rel="canonical" href="https://www.apple.com/" />

<link rel="alternate" href="https://www.apple.com/" hreflang="en-US" /><link rel="alternate" href="https://www.apple.com/ae-ar/" hreflang="ar-AE" /><link rel="alternate" href="https://www.apple.com/ae/" hreflang="en-AE" /><link rel="alternate" href="https://www.apple.com/am/" hreflang="en-AM" /><link rel="alternate" href="https://www.apple.com/at/" hreflang="de-AT" /><link rel="alternate" href="https://www.apple.com/au/" hreflang="en-AU" /><link rel="alternate" href="https://www.apple.com/az/" hreflang="en-AZ" /><link rel="alternate" href="https://www.apple.com/befr/" hreflang="fr-BE" /><link rel="alternate" href="https://www.apple.com/benl/" hreflang="nl-BE" /><link rel="alternate" href="https://www.apple.com/bg/" hreflang="bg-BG" /><link rel="alternate" href="https://www.apple.com/bh-ar/" hreflang="ar-BH" /><link rel="alternate" href="https://www.apple.com/bh/" hreflang="en-BH" /><link rel="alternate" href="https://www.apple.com/br/" hreflang="pt-BR" /><link rel="alternate" href="https://www.apple.com/bw/" hreflang="en-BW" /><link rel="alternate" href="https://www.apple.com/by/" hreflang="en-BY" /><link rel="alternate" href="https://www.apple.com/ca/" hreflang="en-CA" /><link rel="alternate" href="https://www.apple.com/ca/fr/" hreflang="fr-CA" /><link rel="alternate" href="https://www.apple.com/cf/" hreflang="fr-CF" /><link rel="alternate" href="https://www.apple.com/chde/" hreflang="de-CH" /><link rel="alternate" href="https://www.apple.com/chfr/" hreflang="fr-CH" /><link rel="alternate" href="https://www.apple.com/ci/" hreflang="fr-CI" /><link rel="alternate" href="https://www.apple.com/cl/" hreflang="es-CL" /><link rel="alternate" href="https://www.apple.com/cm/" hreflang="fr-CM" /><link rel="alternate" href="https://www.apple.com/co/" hreflang="es-CO" /><link rel="alternate" href="https://www.apple.com/cz/" hreflang="cs-CZ" /><link rel="alternate" href="https://www.apple.com/de/" hreflang="de-DE" /><link rel="alternate" href="https://www.apple.com/dk/" hreflang="da-DK" /><link rel="alternate" href="https://www.apple.com/ee/" hreflang="et-EE" /><link rel="alternate" href="https://www.apple.com/eg-ar/" hreflang="ar-EG" /><link rel="alternate" href="https://www.apple.com/eg/" hreflang="en-EG" /><link rel="alternate" href="https://www.apple.com/es/" hreflang="es-ES" /><link rel="alternate" href="https://www.apple.com/fi/" hreflang="fi-FI" /><link rel="alternate" href="https://www.apple.com/fr/" hreflang="fr-FR" /><link rel="alternate" href="https://www.apple.com/ge/" hreflang="en-GE" /><link rel="alternate" href="https://www.apple.com/gn/" hreflang="fr-GN" /><link rel="alternate" href="https://www.apple.com/gq/" hreflang="fr-GQ" /><link rel="alternate" href="https://www.apple.com/gr/" hreflang="el-GR" /><link rel="alternate" href="https://www.apple.com/gw/" hreflang="en-GW" /><link rel="alternate" href="https://www.apple.com/hk/" hreflang="zh-HK" /><link rel="alternate" href="https://www.apple.com/hk/en/" hreflang="en-HK" /><link rel="alternate" href="https://www.apple.com/hr/" hreflang="hr-HR" /><link rel="alternate" href="https://www.apple.com/hu/" hreflang="hu-HU" /><link rel="alternate" href="https://www.apple.com/id/" hreflang="en-ID" /><link rel="alternate" href="https://www.apple.com/ie/" hreflang="en-IE" /><link rel="alternate" href="https://www.apple.com/il/" hreflang="en-IL" /><link rel="alternate" href="https://www.apple.com/in/" hreflang="en-IN" /><link rel="alternate" href="https://www.apple.com/it/" hreflang="it-IT" /><link rel="alternate" href="https://www.apple.com/jo-ar/" hreflang="ar-JO" /><link rel="alternate" href="https://www.apple.com/jo/" hreflang="en-JO" /><link rel="alternate" href="https://www.apple.com/jp/" hreflang="ja-JP" /><link rel="alternate" href="https://www.apple.com/ke/" hreflang="en-KE" /><link rel="alternate" href="https://www.apple.com/kg/" hreflang="en-KG" /><link rel="alternate" href="https://www.apple.com/kr/" hreflang="ko-KR" /><link rel="alternate" href="https://www.apple.com/kw-ar/" hreflang="ar-KW" /><link rel="alternate" href="https://www.apple.com/kw/" hreflang="en-KW" /><link rel="alternate" href="https://www.apple.com/kz/" hreflang="en-KZ" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-HN" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-AR" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-SV" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-PR" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-DO" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-EC" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-GT" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-PE" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-BO" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-NI" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-VE" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-PA" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-CR" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-UY" /><link rel="alternate" href="https://www.apple.com/la/" hreflang="es-PY" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-AG" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-BS" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-TC" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-TT" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-SR" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-BM" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-BB" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-BZ" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-GD" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-KY" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-LC" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-MS" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-KN" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-AI" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-VC" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-GY" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-VG" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-PR" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-DM" /><link rel="alternate" href="https://www.apple.com/lae/" hreflang="en-JM" /><link rel="alternate" href="https://www.apple.com/li/" hreflang="de-LI" /><link rel="alternate" href="https://www.apple.com/lt/" hreflang="lt-LT" /><link rel="alternate" href="https://www.apple.com/lu/" hreflang="fr-LU" /><link rel="alternate" href="https://www.apple.com/lv/" hreflang="lv-LV" /><link rel="alternate" href="https://www.apple.com/ma/" hreflang="fr-MA" /><link rel="alternate" href="https://www.apple.com/md/" hreflang="en-MD" /><link rel="alternate" href="https://www.apple.com/me/" hreflang="en-ME" /><link rel="alternate" href="https://www.apple.com/mg/" hreflang="en-MG" /><link rel="alternate" href="https://www.apple.com/mk/" hreflang="en-MK" /><link rel="alternate" href="https://www.apple.com/ml/" hreflang="fr-ML" /><link rel="alternate" href="https://www.apple.com/mo/" hreflang="zh-MO" /><link rel="alternate" href="https://www.apple.com/mt/" hreflang="en-MT" /><link rel="alternate" href="https://www.apple.com/mu/" hreflang="fr-MU" /><link rel="alternate" href="https://www.apple.com/mx/" hreflang="es-MX" /><link rel="alternate" href="https://www.apple.com/my/" hreflang="en-MY" /><link rel="alternate" href="https://www.apple.com/mz/" hreflang="en-MZ" /><link rel="alternate" href="https://www.apple.com/ne/" hreflang="fr-NE" /><link rel="alternate" href="https://www.apple.com/ng/" hreflang="en-NG" /><link rel="alternate" href="https://www.apple.com/nl/" hreflang="nl-NL" /><link rel="alternate" href="https://www.apple.com/no/" hreflang="no-NO" /><link rel="alternate" href="https://www.apple.com/nz/" hreflang="en-NZ" /><link rel="alternate" href="https://www.apple.com/om-ar/" hreflang="ar-OM" /><link rel="alternate" href="https://www.apple.com/om/" hreflang="en-OM" /><link rel="alternate" href="https://www.apple.com/ph/" hreflang="en-PH" /><link rel="alternate" href="https://www.apple.com/pl/" hreflang="pl-PL" /><link rel="alternate" href="https://www.apple.com/pt/" hreflang="pt-PT" /><link rel="alternate" href="https://www.apple.com/qa-ar/" hreflang="ar-QA" /><link rel="alternate" href="https://www.apple.com/qa/" hreflang="en-QA" /><link rel="alternate" href="https://www.apple.com/ro/" hreflang="ro-RO" /><link rel="alternate" href="https://www.apple.com/ru/" hreflang="ru-RU" /><link rel="alternate" href="https://www.apple.com/sa-ar/" hreflang="ar-SA" /><link rel="alternate" href="https://www.apple.com/sa/" hreflang="en-SA" /><link rel="alternate" href="https://www.apple.com/se/" hreflang="sv-SE" /><link rel="alternate" href="https://www.apple.com/sg/" hreflang="en-SG" /><link rel="alternate" href="https://www.apple.com/si/" hreflang="en-SI" /><link rel="alternate" href="https://www.apple.com/sk/" hreflang="sk-SK" /><link rel="alternate" href="https://www.apple.com/sn/" hreflang="fr-SN" /><link rel="alternate" href="https://www.apple.com/th/" hreflang="th-TH" /><link rel="alternate" href="https://www.apple.com/tj/" hreflang="en-TJ" /><link rel="alternate" href="https://www.apple.com/tm/" hreflang="en-TM" /><link rel="alternate" href="https://www.apple.com/tn/" hreflang="fr-TN" /><link rel="alternate" href="https://www.apple.com/tr/" hreflang="tr-TR" /><link rel="alternate" href="https://www.apple.com/tw/" hreflang="zh-TW" /><link rel="alternate" href="https://www.apple.com/ua/" hreflang="uk-UA" /><link rel="alternate" href="https://www.apple.com/ug/" hreflang="en-UG" /><link rel="alternate" href="https://www.apple.com/uk/" hreflang="en-GB" /><link rel="alternate" href="https://www.apple.com/uz/" hreflang="en-UZ" /><link rel="alternate" href="https://www.apple.com/vn/" hreflang="en-VN" /><link rel="alternate" href="https://www.apple.com/za/" hreflang="en-ZA" /><link rel="alternate" href="https://www.apple.com.cn/" hreflang="zh-CN" />

Working With an API



NASA Astronomy Picture of the Day (APOD)

API: https://api.nasa.gov/

Working With the APOD API

- Use the demo key DEMO_KEY to get the image of the day

```
let url = URL(string: "https://api.nasa.gov/planetary/apod?date=2005-2-22&api_key=DEMO_KEY")!
Task {
    let (data, response) = try await URLSession.shared.data(from: url)
    if let httpResponse = response as? HTTPURLResponse,
       httpResponse.statusCode == 200,
       let string = String(data: data, encoding: .utf8) {
         print(string)
    }
}
```


```
{
  "date": "2005-02-22",
  "explanation": "Are Saturn's auroras like Earth's?  To help answer this question, the Hubble Space Telescope and the Cassini
spacecraft monitored Saturn's South Pole simultaneously as Cassini closed in on the gas giant in January 2004.  Hubble snapped
images in ultraviolet light, while Cassini recorded radio emissions and monitored the solar wind.  Like on Earth,
…",
  "hdurl": "http://apod.nasa.gov/apod/image/0502/saturnauroras_hst_big.jpg",
  "media_type": "image",
  "service_version": "v1",
  "title": "Persistent Saturnian Auroras",
  "url": "http://apod.nasa.gov/apod/image/0502/saturnauroras_hst.jpg"
}
```

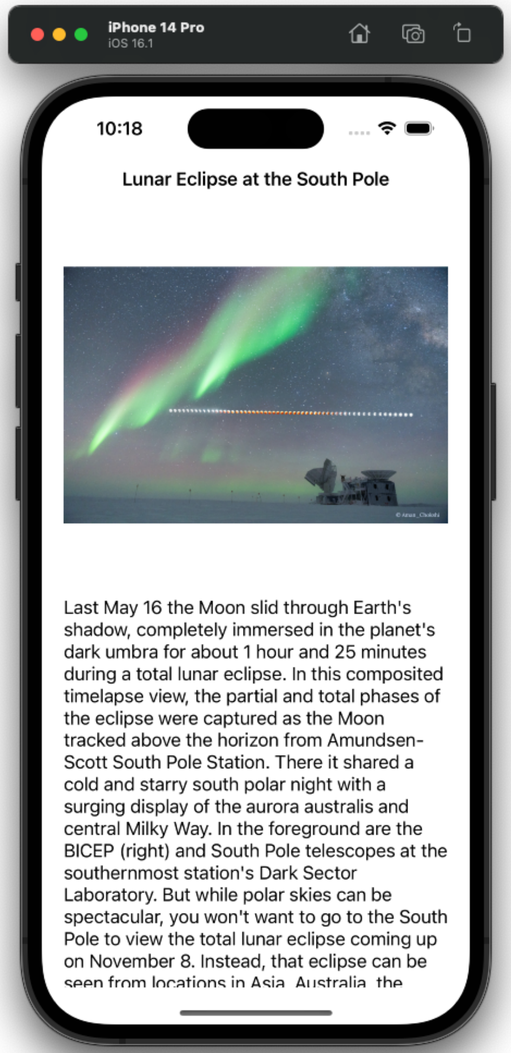
Modify a URL With URL Components

- Dynamically add query items to a URL using URLComponents

```
// Build the URL
var components = URLComponents(string: "https://api.nasa.gov/planetary/apod")!
components.queryItems = [
    "api_key": "DEMO_KEY",
    "date": "2013-07-16"
].map { URLQueryItem(name: $0.key, value: $0.value) }

// Perform the network request
Task {
    let (data, response) = try await URLSession.shared.data(from: components.url!)
    if let httpResponse = response as? HTTPURLResponse,
       httpResponse.statusCode == 200,
       let string = String(data: data, encoding: .utf8) {
         print(string)
    }
}
```

NASA Astronomy Picture of the Day App

1. Create Url ✓
2. Request Data with API keys ✓
3. Create a Swift model
4. Decode JSON
5. Update UI



The Swift Model

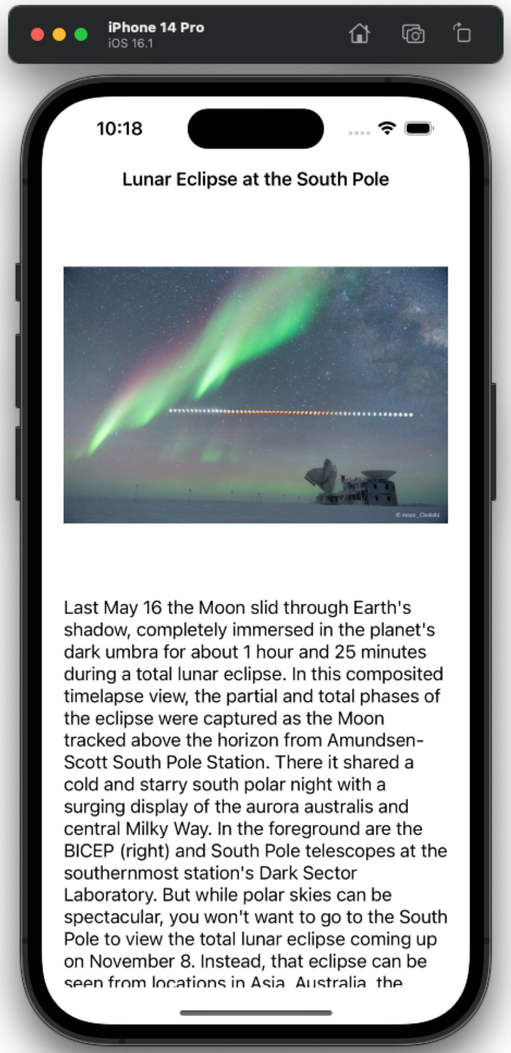
- The PhotoInfo model:

```
{
  "date": "2005-02-22",
  "explanation": "Are Saturn's auroras like Earth's?  To
help answer this question, the Hubble Space Telescope
and the Cassini spacecraft monitored Saturn's South Pole
simultaneously as Cassini closed in on the gas giant in
January 2004.  Hubble snapped images in ultraviolet
light, while Cassini recorded radio emissions and
monitored the solar wind.  Like on Earth,
…",
  "hdurl": "http://apod.nasa.gov/apod/image/0502/
saturnauroras_hst_big.jpg",
  "media_type": "image",
  "service_version": "v1",
  "title": "Persistent Saturnian Auroras",
  "url": "http://apod.nasa.gov/apod/image/0502/
saturnauroras_hst.jpg"
}
```

```
struct PhotoInfo: Codable {
    var title: String
    var description: String
    var url: URL
    var copyright: String?

    enum CodingKeys: String, CodingKey {
        case title
        case description = "explanation"
        case url
        case copyright
    }
}
```

NASA Astronomy Picture of the Day App

1. Create Url ✓
2. Request Data with API keys ✓
3. Create a Swift model ✓
4. Decode JSON
5. Update UI



Decode JSON

```
var components = URLComponents(string: "https://api.nasa.gov/planetary/apod")!
components.queryItems = [
    "api_key": "DEMO_KEY",
    "date": "2013-07-16"
].map { URLQueryItem(name: $0.key, value: $0.value) }

// Perform the network request
Task {
    let (data, response) = try await URLSession.shared.data(from: components.url!)
    let jsonDecoder = JSONDecoder()
    if let httpResponse = response as? HTTPURLResponse,
        httpResponse.statusCode == 200,
        let photoInfo = try? jsonDecoder.decode(PhotoInfo.self, from: data) {
        print(photoInfo)
    }
}
```

Async Calls

```
func fetchPhotoInfo() -> PhotoInfo{
    // Build the URL
    var components = URLComponents(string: "https://api.nasa.gov/planetary/apod")!
    components.queryItems = [
        "api_key": "DEMO_KEY",
        "date": "2013-07-16"
    ].map { URLQueryItem(name: $0.key, value: $0.value) }

    // Perform the network request
    let (data, response) = try await URLSession.shared.data(from: components.url!)
    let jsonDecoder = JSONDecoder()

    if let httpResponse = response as? HTTPURLResponse,
        httpResponse.statusCode == 200,
        let photoInfo = try? jsonDecoder.decode(PhotoInfo.self, from: data) {
        return photoInfo
    }
}
```

'async' call in a function that does not support concurrency

Add 'async' to function 'fetchPhotoInfo()' to make it asynchronous Fix

Errors thrown from here are not handled

Async Calls

```
func fetchPhotoInfo() async throws -> PhotoInfo{
    // Build the URL
    var components = URLComponents(string: "https://api.nasa.gov/planetary/apod")!
    components.queryItems = [
        "api_key": "DEMO_KEY",
        "date": "2013-07-16"
    ].map { URLQueryItem(name: $0.key, value: $0.value) }

    // Perform the network request
    let (data, response) = try await URLSession.shared.data(from: components.url!)
    let jsonDecoder = JSONDecoder()

    if let httpResponse = response as? HTTPURLResponse,
        httpResponse.statusCode == 200,
        let photoInfo = try? jsonDecoder.decode(PhotoInfo.self, from: data) {
        return photoInfo
    }
}
```

Missing return in global function expected to return 'PhotoInfo'

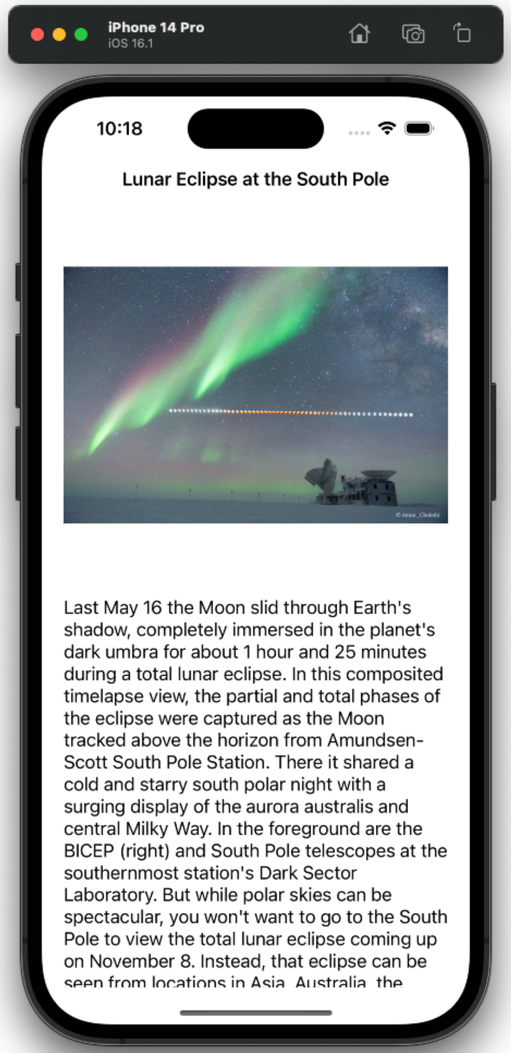
Async Calls

```
enum PhotoInfoError: Error, LocalizedError {
    case itemNotFound
}
```

```
func fetchPhotoInfo() async throws -> PhotoInfo{
    // Buid the URL
    var components = URLComponents(string: "https://api.nasa.gov/planetary/apod")!
    components.queryItems = [
        "api_key": "DEMO_KEY",
        "date": "2013-07-16"
    ].map { URLQueryItem(name: $0.key, value: $0.value) }

    // Perform the network request
    let (data, response) = try await URLSession.shared.data(from: components.url!)

    guard let httpResponse = response as? HTTPURLResponse,
              httpResponse.statusCode == 200 else {
            throw PhotoInfoError.itemNotFound
    }

    let jsonDecoder = JSONDecoder()
    let photoInfo = try jsonDecoder.decode(PhotoInfo.self, from: data)
    return(photoInfo)
}
```

NASA Astronomy Picture of the Day App

1. Create Url ✓
2. Request Data with API keys ✓
3. Create a Swift model ✓
4. Decode JSON ✓
5. Update UI



Update the UI

```
override func viewDidLoad() {
    super.viewDidLoad()

    Task {
        do {
            let photoInfo = try await fetchPhotoInfo()
            updateUI(with: photoInfo)
        } catch {
            updateUI(with: error)
        }
    }
}
```

```
func updateUI(with photoInfo: PhotoInfo) {
    Task {
        do {
            let image = try await fetchImage(from:
                photoInfo.url)
            title = photoInfo.title
            imageView.image = image
            descriptionLabel.text =
                photoInfo.description
            copyrightLabel.text = photoInfo.copyright
        } catch {
            updateUI(with: error)
        }
    }
}
```

Concurrency

Multi Threading in iOS

- Run multiple tasks at the same time
- Run slow or expensive tasks in the background
- Free the main thread so it responds to the UI

Synchronous and Asynchronous

- Synchronous
  - One task completes before another begins
  - Ties up the main thread (main queue)
- Asynchronous
  - Multiple tasks run simultaneously on multiple threads (concurrency)
  - Tasks run in the background thread (background queue)
  - Frees up the main thread

Swift Concurrency

- Swift uses Actors to protect against concurrent updates
- A special Actor called the MainActor is used for UIKit
  - Standard UIKit controllers use the MainActor
  - Safe to update UI in a `Task`'s closure that was created in the context of the MainActor
  - Code after a method that can suspend (marked with await) will run synchronously in the context of the MainActor

Grand Central Dispatch

- Allows your app to execute multiple tasks concurrently on multiple threads
- Assigns tasks to "dispatch queues" and assigns priority
- Controls when your code is executed

Grand Central Dispatch

- Main queue
  - Created when an app launches
  - Highest priority
  - Used to update the UI and respond quickly to user input
- Background queues
  - Lower priority
  - Used to run long-running operations

Dispatch Queue

```
DispatchQueue.global(qos: .background).async {
    // Do some background work
    DispatchQueue.main.async {
        // Update the UI to indicate the work has been completed
    }
}
```