

Photo of the Day (NASA APOD) – VIPER Documentation

1. Overview

โมดูลนี้มีหน้าที่:

- แสดงรายการ NASA Astronomy Picture of the Day (APOD)
- เลือกตาม เดือน / ปี
- แสดงรายการแบบ list
- กดเข้าไปดู รายละเอียดแต่ละรายการ

สถาปัตยกรรมที่ใช้: VIPER

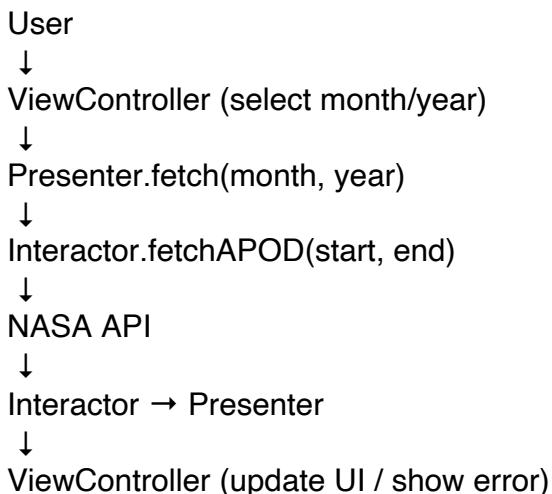
2. Architecture (VIPER)

Responsibility

- View - แสดง UI, รับ user interaction
- Presenter - จัดการ logic, แปลงข้อมูล
- Interactor - business logic, เรียก API
- Entity - Model (APODItem, Error)
- Router - Navigation

3. Data Flow

Load APOD by Month



4. Entity

APODItem

```
struct APODItem Decodable {
    let date String
    let title String
    let explanation String
    let url String
    let mediaType String

    enum CodingKeys String CodingKey {
        case date
        case title
        case explanation
        case mediaType "media_type"
    }
}
```

NASA Error Response

```
struct NASAErrorResponse {
    let code Int
    let msg String
    let service_version String
```

5. Error Handling

APIError

```
enum APIError Error {
    case invalidResponse
    case serverError String
    case decodingError
    case urlError
```

Strategy

- - Decode error body เมื่อ statusCode ≠ 2xx
- - ดึง msg จาก NASA API
- - Presenter แปลงเป็นข้อความที่เหมาะสมกับผู้ใช้

6. Interactor

Fetch APOD

```
fetchAPOD startDate      endDate
```

Responsibility

- - เรียก APIService
- - ไม่รั้งจัก UI
- - ส่งผลลัพธ์กลับผ่าน Output

```
fetchAPOD(startDate: Date, endDate: Date, weak self: PhotoOfDayViewProtocol, in completion: @escaping (Result<[APODItem], Error>) -> Void) {
```

```
    main.async {
```

```
        switch await fetchAPOD(startDate: startDate, endDate: endDate) {
```

```
            case .success(let items):
```

```
                self.output.fetchSuccess(items)
```

```
            case .failure(let error):
```

```
                self.output.fetchFailure(error)
```

7. Presenter

Responsibilities

- แปลง Error → User Message
- ควบคุม View state

```
func fetchFailure(error: Error) {
```

```
    let errorMessage: String
```

```
    view.hideLoading()
```



```
    if case let APIError.serverError = error {
```

```
        errorMessage = "ไม่สามารถโหลดข้อมูลได้ กรุณาลองใหม่"
```



```
        view.showError(errorMessage)
```

8. View

PhotoOfDayViewProtocol

```
protocol PhotoOfDayViewProtocol: AnyObject {
```

```
    func showLoading()
```

```
    func hideLoading()
```

```
    func showPhotos(items: [APODItem])
```

```
    func showError(error: String)
```

UI Components

- - PhotoOfDayViewController
- - PhotoViewer (UIView แยก)
- - PhotoDetailViewController

9. Date Handling

DateHelper

```
private static func gregorianYear from Int Int
static func startOfMonth month Int year Int Date
static func endOfMonth month Int year Int Date
```

Rules

- - endDate ต้องไม่เกิน Date()
- - รองรับเฉพาะช่วง
1995-06-16 → today

10. Global UI Utilities

Global Loading

- - add ลง UIWindow
- - block interaction ทั้งหมด

Global Alert

GlobalAlert show message

ใช้สำหรับ:

- - API error
- - validation error
- - system error

11. Best Practices

- - Interactor ไม่เขี่ย UI
- - Presenter ไม่เรียก UIKit
- - Error ต้อง decode body เสมอ

- - แปลง พ.ศ. ไป ค.ศ. ในตัว
- - ห้ามเลือกเดือนอนาคต

13. Future Improvements

- - `async/await`
- - `pagination` (ถ้า API รองรับ)
- - `caching` (URLCache / disk)
- - `skeleton loading`