

포켓몬 API 활용 앱 만들기 - API 요청

핀 · 2025년 8월 30일

[팔로우](#)

0



PokemonAPI 활용앱 만들기

▼ 목록 보기

1/9



무료 API를 찾아보자

여러 API중에 별도의 절차없이 간단한 GET이 되는 „

포켓몬API

- 간단히 받아올 수 있고 이미지도 포함하고 있음

색상별 포켓몬

```
{  
//... Red에 대한 정보 생략//  
"pokemon_species": [  
 {  
 "name": "charmander",  
 "url": "https://pokeapi.co/api/v2/pokemon-species/4/"  
 },
```

```

{
  "name": "paras",
  "url": "https://pokeapi.co/api/v2/pokemon-species/46/"
},
{
  "name": "charmeleon",
  "url": "https://pokeapi.co/api/v2/pokemon-species/5/"
},
{
  "name": "charizard",
  "url": "https://pokeapi.co/api/v2/pokemon-species/6/"
},
{
  "name": "vileplume",
  "url": "https://pokeapi.co/api/v2/pokemon-species/45/"
},
...

```

<https://pokeapi.co/api/v2/pokemon-color/red>로 요청시 response에 대한 정보다.
붉은색이 포함되어있으면 일단 red요청에 걸리는 모양이다.



다만 문제라하면 여기서는 안에 정보가 많긴하지만 활용할 이미지는 제공하고 있지 않았다. 서식지, 성격, 뭐 등등이 있겠지만, 일단은 시각적으로도 보여주려면 이미지가 있는 앤드포인트가 필요했는데,

<https://pokeapi.co/api/v2/pokemon/25>로 하나의 포켓몬에 접근하면 받아올 수 있었다.

```

"sprites": {
  "back_default": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/back"
  "back_female": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/back"
  "back_shiny": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/back"
  "back_shiny_female": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/back"
  "front_default": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/25"
  "front_female": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/fem"
  "front_shiny": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/shir"
  "front_shiny_female": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/fem"
}
// 세대별
// 게임별

```

25번은 피카츄인데, 응답값은 16000줄이 넘고, 이미지를 나타내는 sprite속성은 200줄을 넘게 포함하고 있었다. 보면 이로치부터 다양한 품들로 인해 세대별.. 게임별.. 암컷 수컷 등등 정말 하나의

포켓몬만해도 다양한 이미지가 존재하는 것이다. 이게 피카츄라서 많은 부분에 다양한 모양새로 등장을 했기때문도 있는듯하다. (콘테스트 공주품까지있다. 3세대 였던가..)



그래서 결국 여기 기본 sprites에서 등장하는 front_default 이미지를 사용할 것 같다. 아니면 female,male정도를 다루는게 보편적인 데이터 구조일 것 같다..! 이로치같은건 나중에 추가하는 걸로...!

요청 로직 추가

일단 Xcode에서도 잘 요청되고 잘 출력되는지 확인하기위해 돌려봤다.

```
func searchAction(_ query: String) async {
    guard let url = URL(string: "\u{baseURL}/\u{query}") else {
        return
    }
    do {
        let (data, _) = try await URLSession.shared.data(from: url)
        result = String(data: data.prefix(1000), encoding: .utf8) ?? "텍스트 변환 실패"
        print(result)
    } catch {
        result = "오류: \u{error}"
    }
}
```

그냥찍으면 너무 길어서 일단 prefix를 뒀는데, 워낙 정보가 많아서 이렇게해서는 사용할수가 없을 것이다..!

일단은 그럼 내가 가져오고자하는 포켓몬의 정보를 골라보자!

사용하고자 하는 것

- 포켓몬 이름 species - name
- 포켓몬 속성 types 배열의 type-name (속성이 1개가 아닌 포켓몬들이 많아서 배열임)
- 이미지(앞,뒤)sprites - [front_default, back_default]

딱이렇게만 하면 될 것 같다. 그러면 일단 입력 받기 위해서 어떤 타입들을 받을지를 알려줘야하니 그 부분부터 정의를 해보자

일단 사용할 포켓몬의 타입을 만들자 ! (불타입 그 타입아님, 개발에서 타입)
당장은 이름, 속성, 이미지 정도만 가져갈 것이다. 성격, 특성, 서식지, 기술 등은 추후에 여유가
되면은 추가하도록하자 (MVP 단계에서는 의도적으로 줄일 필요가 있다.)

```
struct Pokemon: Decodable {
    let id: Int
    let name: String
    let types: [TypeElement]
    let sprites: Sprites
}
```

그리고 일단 decoding 과정에서 필터링할 resultItem 을 정의해줬다.

```
struct SearchResultItem: Identifiable {
    var id: Int
    var name: String
    var types: [String]
    var frontImage: String?
    var backImage: String?
}
```

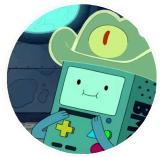
이를 활용해서 searchAction 요청 로직을 변경해주면...!

```
func searchAction(_ query: String) async {
    guard let url = URL(string: "\(baseURL)/\(query)") else {
        return
    }
    do {
        let (data, _) = try await URLSession.shared.data(from: url)
        let pokemon = try JSONDecoder().decode(PokemonResponse.self, from: data)

        let item = SearchResultItem(
            id: pokemon.id,
            name: pokemon.name,
            types: pokemon.types.map { $0.type.name },
            frontImage: pokemon.sprites.front_default,
            backImage: pokemon.sprites.back_default
        )
        print(item)
    } catch {
        result = "오류: \(error)"
        print("에러발생")
    }
}

SearchResultItem(
    id: 1,
    name: "bulbasaur",
    types: ["grass", "poison"],
    frontImage: Optional("https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/1.
```

```
    backImage: Optional("https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/bac  
)
```



팔로우

핀

기억보단 기록을

다음 포스트

포켓몬 API 활용 앱 만들기 - ContentView 분리하기



다운없이 게임 가능!

일본 인기 애니가 게임화, 1200
번 뽑기 무료

0개의 댓글

댓글을 작성하세요

댓글 작성

관심 있을 만한 포스트

다운없이 게임 가능!

일본 인기 애니가 게임화, 1200번 뽑

광고 ansatsu_KR



PASSWORD

.....

TITLE

check me

CONTENT

check my mime type

FILES

파일 선택

1618976872.png

Submit

HTML 폼의 인코딩(enctype), MIME Type(Media Type)

스프링 기반 게시판 애플리케이션을 작성하던 도중 파일이 제대로 전달되지 않는 버그가 있었다.

2021년 4월 27일 · 0개의 댓글



by 하루하음

2

```
je: 'Network Error', name: 'AxiosError', code:  
XMLHttpRequest, ...} i  
)RK"  
:ional: {...}, adapter: Array(2), transformRequest:  
"k Error"  
on"  
)Request {onreadystatechange: null, readyState:  
"or: Network Error\n      at XMLHttpRequest.hand  
Error
```



react-hook-form(Controller) + yup을 활용하여 유효성 검사하기

react-hook-form(Controller) + yup을 활용하여 유효성 검사하기

2023년 10월 11일 · 0개의 댓글



by kim yeseul

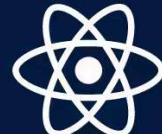
4

Server-Side Rendering (SSR)

with Bun and React



alexkates.dev



[번역] 번(Bun)과 리액트로 서버사이드 렌더링(SSR) 구현하기

번(Bun) 1.0이 출시되며 많은 관심을 받고 있습니다. 이 글은 번을 사용해서 간단한 리액트 SSR 프로젝트를 만들어 볼 수 있는 가이드입니다. 아직 번을 사용해 프로젝트를 만들어보지 못하셨다면 이 가이드를 천천히 따라와보세요:)

React

서버에 form 보내기

[React] 서버에 form 보내기(1)

♥ FormData사용방법과 fetch로 FormData보내기

2022년 9월 15일 · 0개의 댓글

by 혜린

4

[최종 프로젝트 - React with typescript] Supabase(4)_Storage(이미지 여러장 넣고 미리보기)

Supabase Storage 연동 * Supabase Storage의 간단한 구조를 살펴보자면, 웹 페이지에서 업로드한 사진이 storage에 들어가고 그 storage에 들어있는 사진의 url을 원하는 테이블에 지정한 컬럼으로 가져와서 저장한다.

2024년 1월 18일 · 0개의 댓글



by Habin Lee

10

NEXT.JS

[Next.js] Zod 라이브러리 활용하여 데이터 유효성 검사하기

Zod 라이브러리를 활용하여 데이터 유효성을 검사 하는 방법에 대해 알아보자.

2024년 1월 8일 · 0개의 댓글



by 문지은

6



플러터

Flutter



2. 프로젝트 생성 및 위젯

[Flutter] 02. 프로젝트 생성 및 위젯

프로젝트 생성, 간단한 설정, 처음부터 코딩, 위젯, 부모 위젯, 자식 위젯, Text, Icon, Image, asset, Container를 다룹니다.

2024년 2월 3일 · 0개의 댓글



by 김형준

♥ 0



Status: 415 Unsupported Media Type Tim

JSON ▾



Type",

application/x-www-form-urlencoded; charset=UTF-8'

Property List	Dictionary	(15 items)
ion native development re...	String	\$(DEVELOPMENT_LANGUAGE)
ble file	String	\$(EXECUTABLE_NAME)
entifier	String	\$(PRODUCT_BUNDLE_IDENTIFIER)
onary version	String	6.0
ame	String	\$(PRODUCT_NAME)
OS Type code	String	\$(PRODUCT_BUNDLE_PACKAGE_TYPE)
ersion string (short)	String	1.0
ersion	String	1
on requires iPhone enviro...	Boolean	YES
on Scene Manifest	Dictionary	(2 items)
screen interface file base...	String	LaunchScreen
l device capabilities	Array	(1 item)
ed interface orientations	Array	(3 items)
nsport Security Settings	Dictionary	(1 item)
Arbitrary Loads	Boolean	YES
ed interface orientations (i...	Array	(4 items)

iOS : 파운데이션 프레임 워크를 이용한 API 호출

이 글에서 API URL은 전부 생략하였습니다. 라인, 헤더, 바디로 구성. 라인은 http 메시지의 최상단 한줄. 그 다음줄부터는 헤더이며, 헤더가 끝나고 한줄 공백이 있고 그뒤에 나오는게 바디이다. Host 헤더에는 도메인과 포트번호가 적혀있다. 두개 이상의 도메인에 ...

2020년 9월 4일 · 0개의 댓글



by 버들비

1

