//

// DownloadOptionVC.swift

// VideoDownloader

//

// Created by Duy Khanh on 27/09/2023.

//

**import** UIKit

**import** Alamofire

**class** FakeData{

**var** img = ""

**var** name = ""

**var** size = ""

**init**(img: String = "", name: String = "", size: String = "") {

**self**.img = img

**self**.name = name

**self**.size = size

}

}

**class** DownloadOptionVC: BaseViewControler {

**@IBOutlet** **weak** **var** viewBackground: UIView!

**@IBOutlet** **weak** **var** btnDownload: UIButton!

**@IBOutlet** **weak** **var** imgTest: UIImageView!

**@IBOutlet** **weak** **var** lbTest: UILabel!

**var** dataToPass: String?

**var** arr = [Data]()

**var** arrFake = [FakeData]()

**@IBOutlet** **weak** **var** collectView: UICollectionView!

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

setupRadius()

getDataFromApi()

// addBlurEffect()

setupCollView()

fakeData()

}

**func** fakeData(){

**let** i1 = FakeData(img: "avatar", name: "Hehe", size: "30.3MB")

**let** i2 = FakeData(img: "avatar", name: "Hehe", size: "20.3MB")

**let** i3 = FakeData(img: "avatar", name: "Hehe", size: "50.3MB")

arrFake.append(i1)

arrFake.append(i2)

arrFake.append(i3)

collectView.reloadData()

}

**func** getDataFromApi(){

**let** link = dataToPass

**if** **let** url = URL(string: "https://getvidfb.com/downloader") {

startAnimating()

**var** request = URLRequest(url: url)

request.httpMethod = "POST"

**let** postDataString = "link=\(link)&content=&type=public&web=tiktok"

request.httpBody = postDataString.data(using: .utf8)

AF.request(request).responseJSON { [**self**] response **in**

**switch** response.result {

**case** .success(**let** value):

**if** **let** data = value **as**? [String: **Any**] {

**let** univer = data.map({\_ **in** Data.deserialize(from: data)!})

**self**.arr = univer

**self**.collectView.reloadData()

}

// if let apiResponse = ApiResponse.deserialize(from: data) {

// print("Status Code: \(apiResponse.statusCode)")

// for mediaData in apiResponse.media {

// self.stopAnimating()

// print(mediaData.images)

// print(mediaData.videos)

// print(mediaData.mp3s)

// if let imageURLString = mediaData.images as? String {

// let cleanedURLString = imageURLString

// .replacingOccurrences(of: "\n", with: "")

// .replacingOccurrences(of: "\"", with: "")

// .replacingOccurrences(of: "(", with: "")

// .replacingOccurrences(of: ")", with: "")

// .trimmingCharacters(in: .whitespacesAndNewlines)

//

//

// if let imageURL = URL(string: cleanedURLString) {

// let session = URLSession.shared

// let task = session.dataTask(with: imageURL) { (data, response, error) in

// if error == nil, let imageData = data {

// DispatchQueue.main.async {

// if let image = UIImage(data: imageData) {

// self.imgTest.image = image

// self.imgTest.contentMode = .scaleAspectFit

// }

// }

// }

// }

// task.resume()

// }

// }

//

// self.arr.append()

// print("add data")

**case** .failure(\_):

print("Error")

}

}

}

}

**func** setupCollView(){

collectView.dataSource = **self**

collectView.delegate = **self**

**let** nib = UINib(nibName: "CheckBoxCell", bundle: **nil**)

collectView.register(nib, forCellWithReuseIdentifier: "checkBoxCell")

}

**func** setupRadius(){

btnDownload.layer.cornerRadius = 20

btnDownload.layer.masksToBounds = **true**

viewBackground.layer.cornerRadius = 5

viewBackground.layer.masksToBounds = **true**

}

**func** addBlurEffect() {

**let** blurEffect = UIBlurEffect(style: .dark)

**let** blurView = UIVisualEffectView(effect: blurEffect)

blurView.frame = view.bounds

view.addSubview(blurView)

view.sendSubviewToBack(blurView)

view.bringSubviewToFront(viewBackground)

}

**@IBAction** **func** tapOnCancel(\_ sender: **Any**) {

dismiss(animated: **true**)

}

}

**extension** DownloadOptionVC: UICollectionViewDelegate, UICollectionViewDataSource{

**func** collectionView(\_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {

**return** arr.count

}

**func** collectionView(\_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {

**let** cell = collectionView.dequeueReusableCell(withReuseIdentifier: "checkBoxCell", for: indexPath) **as**! CheckBoxCell

**let** mediaData = arr[indexPath.row]

cell.imgAvatar.image = UIImage(named: mediaData.images)

cell.lbSize.text = mediaData.mp3s

**return** cell

}

**func** collectionView(\_ collectionView: UICollectionView, layout collectionViewLayout: UICollectionViewLayout, sizeForItemAt indexPath: IndexPath) -> CGSize {

**return** CGSize(width: 100, height: 100)

//

// DownloadOptionVC.swift

// VideoDownloader

//

// Created by Duy Khanh on 27/09/2023.

//

**import** UIKit

**import** Alamofire

**import** SDWebImage

**class** DownloadOptionVC: BaseViewControler {

**@IBOutlet** **weak** **var** viewBackground: UIView!

**@IBOutlet** **weak** **var** btnDownload: UIButton!

**@IBOutlet** **weak** **var** imgTest: UIImageView!

**@IBOutlet** **weak** **var** lbTest: UILabel!

**var** dataToPass: String?

**var** arr = [Data]()

**@IBOutlet** **weak** **var** collectView: UICollectionView!

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

setupRadius()

getDataFromApi()

// addBlurEffect()

setupCollView()

}

**func** getDataFromApi(){

**let** link = dataToPass

**if** **let** url = URL(string: "https://getvidfb.com/downloader") {

startAnimating()

**var** request = URLRequest(url: url)

request.httpMethod = "POST"

**let** postDataString = "link=\(link)&content=&type=public&web=tiktok"

request.httpBody = postDataString.data(using: .utf8)

AF.request(request).responseJSON { [**self**] response **in**

stopAnimating()

**switch** response.result {

**case** .success(**let** value):

**if** **let** responseData = value **as**? [String: **Any**] {

**if** **let** apiResponse = ApiResponse.deserialize(from: responseData) {

**self**.arr = apiResponse.media

**self**.collectView.reloadData()

}

}

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

stopAnimating()

print("Không lấy được dữ liệu từ sever")

}

}

}

}

**func** setupCollView(){

collectView.dataSource = **self**

collectView.delegate = **self**

**let** nib = UINib(nibName: "CheckBoxCell", bundle: **nil**)

collectView.register(nib, forCellWithReuseIdentifier: "checkBoxCell")

}

**func** setupRadius(){

btnDownload.layer.cornerRadius = 20

btnDownload.layer.masksToBounds = **true**

viewBackground.layer.cornerRadius = 5

viewBackground.layer.masksToBounds = **true**

}

**func** addBlurEffect() {

**let** blurEffect = UIBlurEffect(style: .dark)

**let** blurView = UIVisualEffectView(effect: blurEffect)

blurView.frame = view.bounds

view.addSubview(blurView)

view.sendSubviewToBack(blurView)

view.bringSubviewToFront(viewBackground)

}

**@IBAction** **func** tapOnCancel(\_ sender: **Any**) {

dismiss(animated: **true**)

}

}

**extension** DownloadOptionVC: UICollectionViewDelegate, UICollectionViewDataSource{

**func** collectionView(\_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {

**return** arr.count

}

**func** collectionView(\_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {

**let** cell = collectionView.dequeueReusableCell(withReuseIdentifier: "checkBoxCell", for: indexPath) **as**! CheckBoxCell

**let** mediaData = arr[indexPath.row]

**if** **let** imageURLString = mediaData.images **as**? String {

**let** cleanedURLString = imageURLString

.replacingOccurrences(of: "\n", with: "")

.replacingOccurrences(of: "\"", with: "")

.replacingOccurrences(of: "(", with: "")

.replacingOccurrences(of: ")", with: "")

.trimmingCharacters(in: .whitespacesAndNewlines)

**if** **let** imageURL = URL(string: cleanedURLString) {

**let** session = URLSession.shared

**let** task = session.dataTask(with: imageURL) { (data, response, error) **in**

**if** error == **nil**, **let** imageData = data {

DispatchQueue.main.async {

**if** **let** image = UIImage(data: imageData) {

cell.imgAvatar.sd\_setImage(with: imageURL, placeholderImage: **nil**, options: .refreshCached)

}

}

}

}

task.resume()

}

}

cell.lbSize.text = "\(mediaData.mp3s)"

**return** cell

}

**func** collectionView(\_ collectionView: UICollectionView, layout collectionViewLayout: UICollectionViewLayout, sizeForItemAt indexPath: IndexPath) -> CGSize {

**return** CGSize(width: 100, height: 100)

}

// func collectionView(\_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {

// let mediaData = arr[indexPath.row]

//

// if let imageURLString = mediaData.images as? String{

// // This is an ImagesCell

// let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "imagesCell", for: indexPath) as! ImagesCell

//

// let cleanedURLString = imageURLString

// .replacingOccurrences(of: "\n", with: "")

// .replacingOccurrences(of: "\"", with: "")

// .replacingOccurrences(of: "(", with: "")

// .replacingOccurrences(of: ")", with: "")

// .trimmingCharacters(in: .whitespacesAndNewlines)

// if let imageURL = URL(string: cleanedURLString) {

// let session = URLSession.shared

// let task = session.dataTask(with: imageURL) { (data, response, error) in

// if error == nil, let imageData = data {

// DispatchQueue.main.async {

// if let image = UIImage(data: imageData) {

// cell.imgAvatar.image = image

// }

// }

// } else {

// print("Error loading image: \(error?.localizedDescription ?? "")")

// }

// }

// task.resume()

// }

//

// return cell

// }else if let videoURLString = mediaData.videos as? String {

// // This is a VideosCell

// let cell2 = collectionView.dequeueReusableCell(withReuseIdentifier: "videosCell", for: indexPath) as! VideosCell

// cell2.imgVideos.image = UIImage(named: "avatar")

//

//// let cleanedURLString = videoURLString

//// .replacingOccurrences(of: "\n", with: "")

//// .replacingOccurrences(of: "\"", with: "")

//// .replacingOccurrences(of: "(", with: "")

//// .replacingOccurrences(of: ")", with: "")

//// .trimmingCharacters(in: .whitespacesAndNewlines)

//// if let videoURL = URL(string: "https://v16m-default.akamaized.net/3da718f724ab9627562c57f0d65af3aa/651db09c/video/tos/maliva/tos-maliva-ve-0068c799-us/owXEdlihgzIHQE6IDAfyAsIzAlqbFiNvNlgCwD/?a=0&ch=0&cr=0&dr=0&lr=all&cd=0%7C0%7C0%7C0&cv=1&br=1424&bt=712&bti=OUBzOTg7QGo0NzZAL3AjLTAzYCM1NTNg&cs=0&ds=6&ft=iJOG.y7oZzv0PD1MHM\_xg9wqBSMrBEeC~&mime\_type=video\_mp4&qs=0&rc=OGc4Nzc7OWQ8M2U5MzU8NUBpanF5djY6ZmhlbjMzaTczNEBhMTA2XjNgNl8xXy8vMGEvYSNmLWxmcjRnLy9gLS1kMTJzcw%3D%3D&l=20231004123508BCACED63A358626DFB1C&btag=e00090000") {

//// if let thumbnailImage = getThumbnailImage(forUrl: videoURL) {

//// cell2.imgVideos.image = thumbnailImage

//// }

//// let session = URLSession.shared

//// let task = session.dataTask(with: videoURL) { (data, response, error) in

//// if error == nil, let videoData = data {

//// DispatchQueue.main.async {

//// if let image = UIImage(data: videoData) {

//// cell2.imgVideos.image = image

//// }

//// }

//// } else {

//// print("Error loading image: \(error?.localizedDescription ?? "")")

//// }

//// }

//// task.resume()

//// }

//

//// if let videoURL = URL(string: videoURLString) {

//// // Load a thumbnail image for the video (you need to implement this function)

//// if let thumbnailImage = getThumbnailImage(forUrl: videoURL) {

//// cell2.imgVideos.image = thumbnailImage

//// }

//// }

//

// return cell2

// }

//

// // Return a default cell or handle other cases as needed

// return UICollectionViewCell()

// }

}

}

}