*//*

*// FavoritesEntity.swift*

*// iCatchUpF*

*//*

*// Created by Diego Salas Noain on 7/4/19.*

*// Copyright © 2019 final. All rights reserved.*

*//*

**import** Foundation

**import** CoreData

**import** UIKit

**class** FavoritesEntity {

**let** delegate = UIApplication.shared.delegate **as**! AppDelegate

**let** context = (UIApplication.shared.delegate **as**! AppDelegate).persistentContainer.viewContext

**let** entityName = "Favorite"

**func** add(from source: Source) {

**if** **let** entity = NSEntityDescription.entity(

forEntityName: entityName, in: context) {

**let** newObject = NSManagedObject(entity: entity, insertInto: context)

newObject.setValue( source.id, forKey: "sourceId")

newObject.setValue(source.name, forKey: "sourceName")

print("AGREGADOs")

save()

}

}

**func** save() {

delegate.saveContext()

}

**func** all() -> [NSManagedObject]? {

**return** find(withPredicate: **nil**)

}

**func** find(withPredicate predicate: NSPredicate?) -> [NSManagedObject]? {

**let** request = NSFetchRequest<NSFetchRequestResult>(entityName: entityName)

**if** **let** predicate = predicate {

request.predicate = predicate

}

**do** {

**let** result = **try** context.fetch(request)

**return** result **as**? [NSManagedObject]

} **catch** {

print("Error while querying: \(error.localizedDescription)")

}

**return** **nil**

}

**func** find(byId id: String) -> NSManagedObject? {

**let** predicate = NSPredicate(format: "sourceId = %@", id)

**if** **let** result = find(withPredicate: predicate) {

**return** result.first

}

**return** **nil**

}

**func** find(byName name: String) -> NSManagedObject? {

**let** predicate = NSPredicate(format: "sourceName = %@", name)

**if** **let** result = find(withPredicate: predicate) {

**return** result.first

}

**return** **nil**

}

**func** isFavorite(source: Source) -> Bool {

**return** find(byId: source.id!) != **nil**

}

**func** delete(for source:Source) {

**if** **let** favorite = find(byId: source.id!) {

**do** {

**try** favorite.validateForDelete()

context.delete(favorite)

save()

} **catch** {

print("Error while deleting: \(error.localizedDescription)")

}

}

}

**func** sourceIdsAsString() -> String? {

**if** **let** favorites = all() {

**return** favorites.map({$0.value(forKey: "sourceId") **as**! String}).joined()

}

**return** **nil**

}

}