EBook Gratis

APRENDIZAJE realm

Free unaffiliated eBook created from
Stack Overflow contributors.

#realm

Tabla de contenido

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: realm

It is an unofficial and free realm ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official realm.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con el reino

Observaciones

Realm es una primera base de datos móvil diseñada para proporcionar consultas rápidas con bajo consumo de memoria mediante el uso de la perezosa evaluación, junto con objetos de modelo de tipo seguro y de actualización en vivo.

Realm está disponible para múltiples plataformas y, por lo tanto, abarca varios proyectos:

- Reino Objetivo-C
- Reino veloz
- Reino de Java
- Reino de JavaScript
- Reino xamarin

Versiones

Plataforma	Versión	Fecha de lanzamiento
Reino Objetivo-C	2.4.4	2017-03-14
Reino veloz	2.4.4	2017-03-14
Reino de Java	3.2.0	2017-05-16

Examples

Java

```
public class Dog extends RealmObject {
    public String name;
    public int age;
}

Dog dog = new Dog();
dog.name = "Rex";
dog.age = 1;

Realm realm = Realm.getDefaultInstance();
realm.executeTransaction(new Realm.Transaction() {
    @Override
    public void execute(Realm realm) {
        realm.copyToRealmOrUpdate(dog);
    }
});

RealmResults<Dog> pups = realm.where(Dog.class)
```

```
                                .lessThan("age", 2)
                                .findAll();
```

C objetivo

```
@interface Dog : RLMObject
@property NSString *name;
@property NSInteger age;
@end
@implementation Dog
@end

Dog *dog = [Dog new];
dog.name = @"Rex";
dog.age = 1;

RLMRealm *realm = [RLMRealm defaultRealm];
[realm transactionWithBlock:^{
    [realm addObject:dog];
}];

RLMResults<Dog *> *allDogs = [Dog allObjects];
RLMResults<Dog *> *pups = [allDogs objectsWhere:@"age < 2"];
```

Rápido

Crear clase de objeto:

```
class Dog: Object {
    dynamic var name = ""
    dynamic var age = 0
}
```

Asignar valores de objeto:

```
let dog = Dog()
dog.name = "Rex"
dog.age = 1
```

Guardar objeto:

```
let realm = try! Realm()
try! realm.write {
    realm.add(dog)
}
```

Objetos de lectura:

```
let realm = try! Realm()
let pups = realm.objects(Dog.self)
```

Filtrar objetos:

```
let realm = try! Realm()
let filteredPups = realm.objects(Dog.self).filter("age < 2")
```

Contando objetos:

```
let realm = try! Realm()
let pupsCount = realm.objects(Dog.self).count
```

Xamarin

```
public class Dog : RealmObject
{
    public string Name { get; set; }
    public int Age { get; set; }
}

var realm = Realm.GetInstance();
realm.Write(() =>
{
    var mydog = realm.CreateObject<Dog>();
    mydog.Name = "Rex";
    mydog.Age = 1;
});

var pups = realm.All<Dog>().Where(d => d.Age < 2);
```

Reaccionar-nativo

```
class Dog {}

Dog.schema = {
    name: 'Dog',
    properties: {
        name: 'string',
        age: 'int',
    }
};

let realm = new Realm();
realm.write(() => {
    realm.create('Dog', {name: 'Rex', age: 1});
});

let pups = realm.objects('Dog').filtered('age > 2');
```

Lea Empezando con el reino en línea: https://riptutorial.com/es/realm/topic/1042/empezando-con-el-reino

Capítulo 2: Encuentra la ubicación del archivo Realm

Examples

Imprimir la ubicación del archivo - Swift

Agregue la siguiente línea al método `ViewDidLoad` :

```
print(Realm.Configuration.defaultConfiguration.fileURL!)
```

La línea de arriba imprimirá la ubicación en la consola de Xcode. Copie la ruta del archivo, vaya a **Finder** → Go → Go to Folder ... (o **+ + G**) → pegue la ruta y presione Go.

Ubicación del archivo de impresión - Objective-C

Registre la ubicación del archivo de reino usando:

```
NSLog(@"%@",[RLMRealmConfiguration defaultConfiguration].fileURL);
```

La línea de arriba imprimirá la ubicación en la consola de Xcode. Copie la ruta del archivo, vaya a **Finder** → Go → Go to Folder ... (o **+ + G**) → pegue la ruta y presione Go.

Imprimir la ubicación del archivo - Xamarin

Primero necesitas implementar Realm al comienzo de tu clase.

```
using Realms;
```

Luego para imprimir la ubicación a la consola:

```
Console.WriteLine( RealmConfiguration.PathToRealm() );
```

O si estás usando DefaultConfiguration, puedes usar:

```
Console.WriteLine( RealmConfiguration.DefaultConfiguration.DatabasePath );
```

Cómo llegar al archivo:

Si está ejecutando en **simulador de IOS** :

Puede copiar la ruta del archivo, vaya a **Finder** → Go → Go to Folder ... (o + + G) → pegue la ruta y presione Go.

Pero si estás ejecutando en el **emulador de Android** :

Abra el **monitor del dispositivo Android** (en Visual Studio → menú de herramientas → Android → Monitor del dispositivo Android) (en Xamarin Studio → Menú de herramientas → Abra el monitor del dispositivo Android) → pestaña Explorador de archivos → siga la ruta del archivo

Androide

Copia la base de datos desde el emulador / teléfono para verla. Se puede hacer utilizando ADB:

```
adb pull /data/data/<packagename>/files/
```

Ese comando `Realm.getInstance(getContext())` todos los archivos Realm creados por `Realm.getInstance(getContext())` o `Realm.getInstance(new RealmConfiguration.Builder(context).build())` . El archivo de base de datos predeterminado se llama `default.realm` .

Tenga en cuenta que esto solo funcionará en un emulador o dispositivo rooteado.

Lea Encuentra la ubicación del archivo Realm en línea:
https://riptutorial.com/es/realm/topic/2488/encuentra-la-ubicacion-del-archivo-realm

Capítulo 3: Instalación y configuración

Examples

Rápido

- Prerrequisitos

1. iOS 8 o posterior, macOS 10.9 o posterior, todas las versiones de tvOS y watchOS.
2. Se requiere Xcode 8.0 o posterior. Realm Swift 2.3.0 fue la última versión compatible con Swift 2.xy Xcode 7.3.

- Instalación

    - Marco dinámico

1. Descarga la última versión de Realm y extrae el zip.
2. Vaya a la configuración "General" de su proyecto Xcode. Arrastre `RealmSwift.framework` y `Realm.framework` desde el directorio apropiado de la `Realm.framework` Swift para su proyecto en el directorio `ios/, osx/, tvos/` o `watchos/` a la sección "Binarios incrustados". Asegúrese de seleccionar **Copiar elementos si** es **necesario** (excepto si usa Realm en varias plataformas en su proyecto) y haga clic en **Finalizar** .
3. En la "Configuración de compilación" del objetivo de prueba de su unidad, agregue la ruta principal a `RealmSwift.framework` en la sección "Rutas de búsqueda de marco".
4. Si usa Realm en un proyecto de iOS, tvOS o watchOS, cree una nueva "Fase de guión de ejecución" en las "Fases de creación" del objetivo de su aplicación y pegue el siguiente fragmento de código en el campo de texto del guión:

```
  bash "${BUILT_PRODUCTS_DIR}/${FRAMEWORKS_FOLDER_PATH}/Realm.framework/strip-
frameworks.sh"
```

Este paso es necesario para solucionar un error de envío de App Store al archivar binarios universales.

    - CocoaPods

1. Instala CocoaPods 0.39.0 o posterior. .
2. Ejecute la `pod repo update` para que CocoaPods esté al tanto de las últimas versiones de Realm disponibles.
3. En tu Podfile, agrega `use_frameworks!` y `pod 'RealmSwift'` a sus objetivos principales y de prueba.
4. Si usa Xcode 8, pegue lo siguiente en la parte inferior de su Podfile, actualizando la versión Swift si es necesario:

```
post_install do |installer|
  installer.pods_project.targets.each do |target|
    target.build_configurations.each do |config|
      config.build_settings['SWIFT_VERSION'] = '3.0'
    end
```

```
  end
end
```

5. Desde la línea de comandos, ejecute `pod install` .
6. ¡Utilice el archivo `.xcworkspace` generado por CocoaPods para trabajar en su proyecto!

◦ Cartago

1. Instale Cartago 0.17.0 o posterior.
2. Agregue `github "realm/realm-cocoa"` a su Cartfile.
3. Ejecutar la `carthage update` .
4. Arrastre `RealmSwift.framework` y `Realm.framework` desde el directorio de la plataforma correspondiente en `Carthage/Build/` a la sección "Marcos y bibliotecas vinculadas" de la configuración "General" de su proyecto Xcode.
5. **iOS / tvOS / watchOS** : En la pestaña de configuración de "Fases de creación" de los objetivos de la aplicación, haga clic en el icono "+" y seleccione "Nueva fase de secuencia de comandos de ejecución". Crea un script de ejecución con los siguientes contenidos:

```
/usr/local/bin/carthage copy-frameworks
```

y agregue las rutas a los marcos que desea usar en "Archivos de entrada", por ejemplo:

```
$(SRCROOT)/Carthage/Build/iOS/Realm.framework
$(SRCROOT)/Carthage/Build/iOS/RealmSwift.framework
```

Este script funciona en torno a un error de envío de la App Store activado por binarios universales.

Reaccionar-nativo

- Prerrequisitos
- Instalación

Xamarin

- Prerrequisitos
- Instalación

C objetivo

- Prerrequisitos

1. iOS 7 o posterior, macOS 10.9 o posterior, todas las versiones de tvOS y watchOS.
2. Se requiere Xcode 7.3 o posterior.

- Instalación

1. Descargue la última versión de los archivos Realm desde aquí o desde el enlace de Github y extraiga el archivo zip.
2. Navegue al directorio ios / static /
3. Arrastre Realm.framework al navegador de archivos de su proyecto Xcode. Asegúrese de seleccionar Copiar elementos si es necesario y haga clic en Finalizar.
4. Haga clic en su proyecto en el navegador de archivos Xcode. Seleccione el destino de su aplicación y vaya a la pestaña Construir fases. Bajo Link Binary with Libraries, haga clic en + y agregue libc ++. Tbd.

Java

- Prerrequisitos
- Instalación

Lea Instalación y configuración en línea: https://riptutorial.com/es/realm/topic/1521/instalacion-y-configuracion

Creditos

S. No	Capítulos	Contributors
1	Empezando con el reino	Community, EpicPandaForce, Marc, sangjoon moon, Sergey, ZGski
2	Encuentra la ubicación del archivo Realm	Agung Santoso, Idan, MujtabaFR
3	Instalación y configuración	EpicPandaForce, Prav, sangjoon moon