

 eBook Gratuit

APPRENEZ realm

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#realm

Table des matières

À propos	1
Chapitre 1: Démarrer avec le royaume	2
Remarques.....	2
Versions.....	2
Exemples.....	2
Java.....	2
Objectif c.....	3
Rapide.....	3
Xamarin.....	4
React-Native.....	4
Chapitre 2: Installation et configuration	5
Exemples.....	5
Rapide.....	5
React-Native.....	6
Xamarin.....	6
Objectif c.....	6
Java.....	7
Chapitre 3: Rechercher l'emplacement du fichier Realm	8
Exemples.....	8
Emplacement du fichier d'impression - Swift.....	8
Emplacement du fichier d'impression - Objective-C.....	8
Emplacement du fichier d'impression - Xamarin.....	8
Comment accéder au fichier:.....	8
Android.....	9
Crédits	10

À propos

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.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

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

Chapitre 1: Démarrer avec le royaume

Remarques

Realm est une solution de base de données mobile conçue pour fournir des requêtes rapides à faible consommation de mémoire via une évaluation différée, ainsi que des objets de modèle à mise à jour sécurisée et en direct.

Le domaine est disponible pour plusieurs plates-formes et englobe donc plusieurs projets:

- [Realm Objective-C](#)
- [Royaume Swift](#)
- [Royaume Java](#)
- [Realm JavaScript](#)
- [Royaume Xamarin](#)

Versions

Plate-forme	Version	Date de sortie
Realm Objective-C	2.4.4	2017-03-14
Royaume Swift	2.4.4	2017-03-14
Royaume Java	3.2.0	2017-05-16

Exemples

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();
```

Objectif c

```
@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"];
```

Rapide

Créer une classe d'objet:

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

Attribuer des valeurs d'objet:

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

Enregistrer l'objet:

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

Objets de lecture:

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

Filtrage des objets:

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

Comptage d'objets:

```
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);
```

React-Native

```
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');
```

Lire Démarrer avec le royaume en ligne: <https://riptutorial.com/fr/realm/topic/1042/demarrer-avec-le-royaume>

Chapitre 2: Installation et configuration

Exemples

Rapide

- [Conditions préalables](#)

1. iOS 8 ou version ultérieure, macOS 10.9 ou version ultérieure, toutes les versions de tvOS et watchOS.
2. Xcode 8.0 ou ultérieur requis. [Realm Swift 2.3.0](#) était la dernière version à prendre en charge Swift 2.x et Xcode 7.3.

- [Installation](#)

- Cadre dynamique

1. Téléchargez [la dernière version de Realm](#) et extrayez le zip.
2. Accédez aux paramètres généraux de votre projet Xcode. Faites glisser `RealmSwift.framework` et `Realm.framework` du `Realm.framework` `Swift-versioned` correspondant à votre projet dans le répertoire `ios/`, `osx/`, `tvos/` ou `watchos/` vers la section «Embedded Binaries». Assurez-vous que l'**option Copier les éléments si nécessaire** est sélectionnée (sauf si vous utilisez Realm sur plusieurs plates-formes de votre projet) et cliquez sur **Terminer**.
3. Dans les «Paramètres de construction» de la cible de test de votre unité, ajoutez le chemin parent à `RealmSwift.framework` dans la section «Chemins de recherche du cadre».
4. Si vous utilisez Realm dans un projet iOS, tvOS ou watchOS, créez une nouvelle «phase de script» dans les «phases de construction» de la cible de votre application et collez l'extrait de code suivant dans le champ de texte du script:

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

Cette étape est nécessaire pour contourner un [bogue de soumission App Store](#) lors de l'archivage de fichiers binaires universels.

- CocoaPods

1. [Installez CocoaPods 0.39.0 ou ultérieur](#).
2. Exécutez la `pod repo update` à `pod repo update` du référentiel de `pod repo update` pour que CocoaPods connaisse les dernières versions de Realm disponibles.
3. Dans votre Podfile, ajoutez `use_frameworks!` et `pod 'RealmSwift'` vers vos cibles principales et vos cibles de test.
4. Si vous utilisez Xcode 8, collez ce qui suit au bas de votre Podfile, en mettant à jour la version de Swift si nécessaire:

```
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
end
```

5. À partir de la ligne de commande, exécutez l' `pod install` .
6. Utilisez le fichier `.xcworkspace` généré par CocoaPods pour travailler sur votre projet!
 - Carthage
 1. [Installez Carthage 0.17.0 ou supérieur](#).
 2. Ajoutez `github "realm/realm-cocoa" cocoa github "realm/realm-cocoa"` à votre Cartfile.
 3. Exécutez la `carthage update` .
 4. Faites glisser `RealmSwift.framework` et `Realm.framework` du `Realm.framework` de plate-forme approprié dans `Carthage/Build/` vers la section «Frameworks et bibliothèques liés» des paramètres «généraux» de votre projet Xcode.
 5. **iOS / tvOS / watchOS** : Dans l'onglet des paramètres de votre application "Build Phases", cliquez sur l'icône "+" et choisissez "New Run Script Phase". Créez un script d'exécution avec le contenu suivant:

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

et ajoutez les chemins d'accès aux frameworks que vous souhaitez utiliser sous "Fichiers d'entrée", par exemple:

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

Ce script fonctionne autour d'un [bogue de soumission App Store](#) déclenché par des binaires universels.

React-Native

- [Conditions préalables](#)
- [Installation](#)

Xamarin

- [Conditions préalables](#)
- [Installation](#)

Objectif c

- [Conditions préalables](#)
1. iOS 7 ou version ultérieure, macOS 10.9 ou version ultérieure, toutes les versions de tvOS et watchOS.

2. Xcode 7.3 ou ultérieur requis.

- [Installation](#)

1. Téléchargez la dernière version des fichiers Realm [ici](#) ou à partir du [lien Github](#) et extrayez le fichier zip.
2. Naviguez vers le répertoire ios / static /
3. Faites glisser Realm.framework vers le navigateur de fichiers de votre projet Xcode. Assurez-vous que l'option Copier les éléments est sélectionnée et cliquez sur Terminer.
4. Cliquez sur votre projet dans le navigateur de fichiers Xcode. Sélectionnez la cible de votre application et accédez à l'onglet Build Phases. Sous Link Binary with Libraries, cliquez sur + et ajoutez libc ++. Tbd.

Java

- [Conditions préalables](#)
- [Installation](#)

Lire [Installation et configuration en ligne](https://riptutorial.com/fr/realms/topic/1521/installation-et-configuration): <https://riptutorial.com/fr/realms/topic/1521/installation-et-configuration>

Chapitre 3: Rechercher l'emplacement du fichier Realm

Exemples

Emplacement du fichier d'impression - Swift

Ajoutez la ligne suivante à la méthode `ViewDidLoad` :

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

La ligne ci-dessus imprimera l'emplacement sur la console de Xcode. Copiez le chemin du fichier, allez dans le **Finder** → Aller → Aller au dossier ... (ou **+ + G**) → collez le chemin et appuyez sur OK.

Emplacement du fichier d'impression - Objective-C

Consignez l'emplacement du fichier de domaine à l'aide de:

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

La ligne ci-dessus imprimera l'emplacement sur la console de Xcode. Copiez le chemin du fichier, allez dans le **Finder** → Aller → Aller au dossier ... (ou **+ + G**) → collez le chemin et appuyez sur OK.

Emplacement du fichier d'impression - Xamarin

Tout d'abord, vous devez implémenter Realm au début de votre cours.

```
using Realms;
```

Ensuite, pour imprimer l'emplacement sur la console:

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

Ou si vous utilisez `DefaultConfiguration`, vous pouvez utiliser:

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

Comment accéder au fichier:

Si vous utilisez le **simulateur IOS** :

Vous pouvez copier le chemin du fichier, allez dans le **Finder** → Aller → Aller au dossier ... (ou + + G) → collez le chemin et appuyez sur OK.

Mais si vous utilisez un **émulateur Android** :

Ouvrez le **moniteur de périphérique Android** (sur Visual Studio → menu Outils → Android → Moniteur de périphérique Android) (sur le studio Xamarin → Menu Outils → Ouvrir le moniteur de périphérique Android) → onglet Explorateur de fichiers → suivez le chemin du fichier

Android

Copiez la base de données de l'émulateur / téléphone pour l'afficher. Cela peut être fait en utilisant ADB:

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

Cette commande `Realm.getInstance(getContext())` tous les fichiers Realm créés par `Realm.getInstance(getContext())` **OU** `Realm.getInstance(new RealmConfiguration.Builder(context).build())` . Le fichier de base de données par défaut s'appelle `default.realm` .

Notez que cela ne fonctionnera que sur un émulateur ou un périphérique rooté.

Lire [Rechercher l'emplacement du fichier Realm en ligne](https://riptutorial.com/fr/realm/topic/2488/rechercher-l-emplacement-du-fichier-realm):

<https://riptutorial.com/fr/realm/topic/2488/rechercher-l-emplacement-du-fichier-realm>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec le royaume	Community , EpicPandaForce , Marc , sangjoon moon , Sergey , ZGski
2	Installation et configuration	EpicPandaForce , Prav , sangjoon moon
3	Rechercher l'emplacement du fichier Realm	Agung Santoso , Idan , MujtabaFR